

Indice analitico

Glossario dei termini principali

AIE2 (*Aritmetica Inversiva Etichettata/Estesa*): teoria che estende l'aritmetica classica introducendo infiniti etichettati ∞_n , un indeterminato $\perp$ e regole operative coerenti.

Infinito etichettato Oggetto $\infty_n = [n, n + 1)$ che rappresenta l'intervallo dei numeri compresi tra n e $n + 1$, con etichetta n che ne identifica la posizione.

Indeterminato L'oggetto $\perp := \frac{0}{0}$, che non ha valore definito e si propaga come caso speciale nelle operazioni.

Forma Normale (FN) Rappresentazione canonica di ogni espressione AIE2 come unione disgiunta e ordinata di intervalli $[a, b)$.

Regolatore R Funzione che prende una FN e la trasforma in una forma regolarizzata (ad esempio numerica o simbolica).

Valutatore Φ Funzione che, a partire dal risultato di R , restituisce un valore in un dato dominio di interesse (numerico, fisico, economico, computazionale).

Somma Minkowskiana L'operatore $\oplus$ definito tra intervalli come $[a, b) \oplus [c, d) = [a + c, b + d)$.

Prodotto di intervalli L'operatore $\odot$ definito come $[a, b) \odot [c, d) = [ac, bd)$ nel caso positivo, con estensione generale a segni diversi.

Scalatura Operatore $k \cdot [a, b) = [ka, kb)$, con $k \in \mathbb{N}$.

Elenco dei simboli

Simbolo	Significato
---------	-------------

∞_n	Infinito etichettato, intervallo unitario $[n, n + 1)$.
------------	--

$\perp$	Indeterminato, definito come $0/0$.
---------	--------------------------------------

$\oplus$	Somma Minkowskiana di intervalli.
----------	-----------------------------------

$\odot$	Prodotto di intervalli.
---------	-------------------------

$k \cdot$	Scalatura di intervalli.
-----------	--------------------------

4. $k \cdot \infty_n = [kn, kn + k) = \bigcup_{i=0}^{k-1} \infty_{kn+i}$.
5. $\infty_m \odot \infty_n = [mn, (m+1)(n+1)) = \bigcup_{i=0}^{m+n} \infty_{mn+i}$.
6. Propagazione di $\perp$: qualsiasi operazione che coinvolge $\perp$ restituisce $\perp$.
7. Forma Normale: ogni unione finita di intervalli è riscrivibile come $[p, q)$.

Come usare l'indice analitico

Questo indice ha lo scopo di:

- fornire un accesso rapido ai concetti base di AIE2;
- ricordare le regole operative in forma sintetica;
- aiutare il lettore a orientarsi nella simbologia del libro senza dover tornare ogni volta ai capitoli introduttivi.

Questo indice, insieme al glossario e alla bibliografia, rende il volume un manuale di riferimento utilizzabile anche a consultazione rapida.

AIE2

Aritmetica Inversiva Etichettata/Estesa

Un nuovo framework per infinito e indeterminato

Salvatore Crapanzano

*A chi guarda l'infinito non come un limite,
ma come una possibilità.*

August 24, 2025

Edizione 2025

© 2025 Salvatore Crapanzano. Tutti i diritti riservati.

Questo volume è distribuito con licenza **Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)**.

Ciò significa che sei libero di:

- condividere — copiare e ridistribuire il materiale in qualsiasi formato o supporto;
- adattare — remixare, trasformare e sviluppare il materiale per qualsiasi scopo, anche commerciale.

A condizione che:

- venga riconosciuta adeguata attribuzione agli autori;
- le eventuali opere derivate siano distribuite con la stessa licenza.

Per il testo completo della licenza:

AIEdred<https://creativecommons.org/licenses/by-sa/4.0/>

Chapter 1

Introduzione alla Teoria AIE2

Riassunto

In sintesi: AIE2 (Aritmetica Inversiva Etichettata/Estesa) è un'estensione coerente dell'aritmetica che attribuisce *valore informativo* alle espressioni singolari come la divisione per zero. Invece di un unico simbolo di infinito, introduce gli *infiniti etichettati* ∞_n con interpretazione intervallare $\infty_n = [n, n + 1)$. Questo permette di eseguire calcoli mantenendo memoria del numeratore e di trattare divergenze in modo strutturato.

1.1 Perché nasce AIE2

L'aritmetica classica lascia *non definita* l'espressione $\frac{n}{0}$ perché non esiste alcun reale x tale che $0 \cdot x = n$ per $n \neq 0$. In molte discipline (fisica teorica, calcolo simbolico, informatica numerica, economia matematica) questa esclusione crea un *vuoto operativo*: le formule che modellano fenomeni reali possono generare *singolarità* (divisioni per zero, divergenze, limiti infiniti) proprio dove la teoria classica non consente di proseguire il ragionamento.

AIE2 nasce per colmare questo vuoto con due idee chiave:

I.1 Estensione controllata del dominio: quando si presenta $\frac{n}{0}$, non si “ferma” il calcolo, ma si passa a un dominio esteso che conserva informazione sul numeratore.

I.2 Etichettatura degli infiniti: a ciascun numeratore $n \neq 0$ si associa un *infinito etichettato* ∞_n , non confondibile con ∞_m se $m \neq n$. L'interpretazione *numerica* è intervallare:

$$\infty_n[n, n + 1) \subset \mathbb{R}.$$

Intuizione

Intuizione rapida. In AIE2, $\frac{1}{0}$ non è un singolo simbolo “ ∞ ”, bensì *tutti* i valori reali tra 1 incluso e 2 escluso: $\infty_1 = [1, 2)$. Analogamente, $\frac{2}{0} = \infty_2 = [2, 3)$, e così via.

Questa scelta rende possibile:

- **continuare i calcoli** oltre le singolarità, mantenendo memoria del numeratore;
- **stimare e controllare** la propagazione delle “zone di indeterminazione” (intervalli) anziché collassare tutto in un simbolo unico privo di struttura;
- **ricondere** molte divergenze a operazioni su insiemi numerici concreti (intervalli), più facili da visualizzare, comporre, approssimare e verificare.

Attenzione

Attenzione. ∞_n non è l’ ∞ dell’analisi classica e non è un numero reale: è una *classe di valori* (un intervallo). Le regole operative tra ∞_n saranno quindi definite *come operazioni tra insiemi* (somma di Minkowski, prodotto puntuale, scalatura), e verranno formalizzate nei capitoli successivi.

Riassunto

Punti chiave della sezione ??:

1. La divisione per zero è trattata in un dominio esteso, non eliminata.
2. A ciascun $n \neq 0$ si associa $\infty_n = [n, n + 1)$.
3. L’informazione sul numeratore è preservata e “vive” nell’etichetta n .

1.2 Differenze con l’aritmetica classica

Confrontiamo i principi operativi.

(A) Divisione per zero.

- **Classico:** $\frac{n}{0}$ non è definito (si interrompe il calcolo).

- **AIE2:** $\frac{n}{0} = \infty_n$ con $\infty_n = [n, n+1)$; il calcolo prosegue nel dominio esteso.

(B) Informazione sul numeratore.

- **Classico:** qualunque “spinta all’infinito” è indistinta (un unico ∞ concettuale).
- **AIE2:** gli “infiniti” sono *etichettati*; $\infty_1 \neq \infty_2$ perché $[1, 2) \neq [2, 3)$.

(C) Natura dell’oggetto infinito.

- **Classico:** l’infinito non è un numero; spesso è gestito per via topologica (punto all’infinito, limiti).
- **AIE2:** ∞_n è un *insieme numerico concreto* (intervallo), su cui definire operazioni insiemistiche.

(D) Operatività.

- **Classico:** si evita di manipolare direttamente espressioni come $1/0$.
- **AIE2:** si manipolano ∞_n con regole precise (cap. ??), mantenendo tracciabilità.

Esempio 1.1. *Esempio motivante.* In presenza di due singolarità, la somma di Minkowski restituisce un *nuovo insieme*:

$$\infty_1 + \infty_1 = [1, 2) + [1, 2) = [2, 4) = \infty_2 \cup \infty_3.$$

Questo evidenzia come il risultato *non* collassi ad un unico simbolo indifferenziato, ma mantenga struttura.

Riassunto

Punti chiave della sezione ??:

- AIE2 sostituisce il “blocco” della non-definizione con un’estensione *strutturata*.
- Gli infiniti diventano *intervalli etichettati* e sono manipolabili come insiemi numerici.

1.3 Obiettivi e campo di applicazione

Gli obiettivi di AIE2 sono tre:

- O.1 Rigore:** definire un sistema assiomatico chiaro (cap. ??) che regoli gli oggetti estesi.
- O.2 Operatività:** fornire regole di calcolo *effettive* tra infiniti etichettati (cap. ??).
- O.3 Utilità:** mostrare applicazioni dove le divergenze diventano gestibili e informativamente utili (parte ??).

Fisica

- **Regolarizzazione di divergenze:** le catastrofi (es. ultravioletta) possono essere trattate come operazioni su unioni di intervalli, introducendo *criteri di selezione/compressione* (cap. applicazioni di fisica).
- **Singularità gravitazionali:** separare contributi “infiniti” in classi ∞_n consente analisi più fini.

Informatica

- **Overflow/underflow concettuale:** invece di saturare a $\pm\infty$ o a “NaN”, si propaga un oggetto strutturato che preserva informazione (*fault-tolerance semantica*).
- **Calcolo simbolico:** regole algoritmiche su intervalli e loro unioni, con tracciabilità delle etichette.

Economia e scienze sociali

- **Modelli con divergenze:** le esplosioni di prezzo/volatilità si mappano su classi ∞_n , consentendo confronti tra scenari infiniti distinti.

- **Analisi di sensitività:** la granularità per etichette n facilita scenari “stressati” ma interpretabili.

Riassunto

Punti chiave della sezione ??:

- Obiettivo triplice: rigore, operatività, utilità applicativa.
- AIE2 rende *computabili* (e verificabili) molte situazioni che la teoria classica lascia fuori dal dominio.

1.4 Come leggere il libro

Il testo è organizzato per consentire due livelli di lettura:

1. **Linea concettuale** (cap. ??–??): motivazioni, idee, esempi guidati.
2. **Linea formale** (cap. ?? e appendici): assiomi, teoremi, dimostrazioni, formalizzazione in Lean/Coq.

Ogni capitolo termina con un *riassunto operativo* e richiami puntuali alle definizioni essenziali da ricordare.

Riassunto

Cosa ti porti via dal Capitolo ??:

1. La *ragione d'essere* di AIE2: estendere l'aritmetica per trattare singolarità con struttura informativa.
2. L'*idea centrale*: $\frac{n}{0} = \infty_n$ con $\infty_n = [n, n + 1)$.
3. Il *programma del libro*: dagli assiomi alle applicazioni, passando per regole operative ed esempi.

Chapter 2

Notazioni e Simboli

Riassunto

In sintesi: in questo capitolo fissiamo la notazione ufficiale di AIE2. Ogni simbolo viene presentato con definizione, interpretazione intuitiva ed esempi. Questo rende il linguaggio coerente in tutto il libro.

2.1 Convenzioni generali

- L'insieme dei numeri naturali è denotato con $\mathbb{N} = \{0, 1, 2, \dots\}$.
- L'insieme dei numeri interi con $\mathbb{Z}$, dei razionali con $\mathbb{Q}$, dei reali con $\mathbb{R}$.
- L'operatore di intervallo semiaperto è scritto come $[a, b)$, che rappresenta tutti i reali x con $a \leq x < b$.
- La scrittura “=” tra insiemi indica uguaglianza insiemistica (stessi elementi).
- Per “ $\cup$ ” intendiamo unione di insiemi, per “ $\cap$ ” intersezione.

Attenzione

In AIE2 useremo intervalli semiaperti $[n, n+1)$ come unità fondamentali. È importante distinguere: non sono punti singolari, ma insiemi di infiniti valori reali.

2.2 Simboli fondamentali

1. Infinito etichettato

Definizione 2.1. Per ogni $n \in \mathbb{Z} \setminus \{0\}$ definiamo

$$\frac{n}{0} := \infty_n, \quad \text{dove} \quad \infty_n = [n, n+1).$$

Intuizione

$\infty_1 = [1, 2)$ è l'insieme di tutti i numeri reali compresi tra 1 incluso e 2 escluso. Analogamente, $\infty_2 = [2, 3)$, $\infty_3 = [3, 4)$, e così via. Questi intervalli partizionano $[1, \infty)$ in blocchi unitari etichettati.

2. Indeterminato

Definizione 2.2. Il caso $\frac{0}{0}$ è denotato da

$$\frac{0}{0} := \perp,$$

dove $\perp$ rappresenta l'*indeterminato*, un oggetto che non appartiene a nessun intervallo ∞_n .

Attenzione

$\perp$ non è un intervallo, non è calcolabile con le stesse regole di ∞_n . È un oggetto distinto che verrà trattato con regole apposite (cap. ??).

3. Operazioni sugli infiniti etichettati

- **Somma:** $\infty_m + \infty_n$ è definita come somma di Minkowski degli intervalli $[m, m+1)$ e $[n, n+1)$.
- **Prodotto:** $\infty_m \cdot \infty_n$ è definito come insieme dei prodotti ab con $a \in \infty_m, b \in \infty_n$.
- **Scalatura:** $k \cdot \infty_n$ con $k \in \mathbb{N}$ è l'intervallo dei multipli ka con $a \in \infty_n$.

Queste regole operative verranno formalizzate nel cap. ??.

2.3 Glossario dei termini

Infinito etichettato ∞_n :

intervallo $[n, n + 1)$ associato al rapporto $\frac{n}{0}$ con $n \neq 0$.

Indeterminato $\perp$:

simbolo che rappresenta $\frac{0}{0}$, non riducibile ad alcun intervallo.

Somma di Minkowski:

per insiemi $A, B \subset \mathbb{R}$, $A + B = \{a + b : a \in A, b \in B\}$.

Prodotto insiemistico:

per insiemi $A, B \subset \mathbb{R}$, $A \cdot B = \{ab : a \in A, b \in B\}$.

Partizione degli infiniti:

unione disgiunta $\bigcup_{n=1}^{\infty} \infty_n = [1, \infty)$.

Esempio 2.1.

$$\infty_1 = [1, 2)$$

$$\infty_2 = [2, 3)$$

$$\infty_1 + \infty_2 = [1, 2) + [2, 3) = [3, 5) = \infty_3 \cup \infty_4.$$

2.4 Riassunto del capitolo

Riassunto

Cosa ricordare:

1. $\frac{n}{0} = \infty_n$ per $n \neq 0$, con $\infty_n = [n, n + 1)$.
2. $\frac{0}{0} = \perp$, simbolo distinto da ogni ∞_n .
3. Le operazioni (somma, prodotto, scalatura) sono definite come operazioni insiemistiche.
4. Gli ∞_n formano una partizione di $[1, \infty)$ in intervalli unitari.

Chapter 3

Sistema Assiomatico di AIE2

Riassunto

In sintesi: fissiamo il *dominio esteso* di AIE2, introduciamo gli *infiniti etichettati* $\infty_n = [n, n+1)$ (per $n \in \mathbb{N}$, $n \geq 1$), l'*indeterminato* $\perp$ per il caso $0/0$, e definiamo le operazioni primitive come *operazioni insiemistiche*: somma di Minkowski, prodotto puntuale e scalatura (per scalari interi positivi). Forniamo inoltre le regole di propagazione di $\perp$ e la *forma normale* per le unioni finite di intervalli etichettati.

3.1 Dominio esteso e oggetti primitivi

Definizione 3.1 (Dominio esteso AIE2). Sia $\mathcal{I}\{\infty_n : n \in \mathbb{N}, n \geq 1\}$ con $\infty_n[n, n+1) \subset \mathbb{R}$. Sia $\mathcal{U}$ l'insieme di *tutte e sole* le **unioni finite di intervalli consecutivi** presi da $\mathcal{I}$, ossia ogni $U \in \mathcal{U}$ ammette una rappresentazione del tipo

$$U = \bigcup_{k=a}^b \infty_k = [a, b+1) \quad \text{per qualche} \quad 1 \leq a \leq b, \quad a, b \in \mathbb{N}.$$

Definiamo il **dominio esteso**

$$\mathbb{A}\mathcal{U} \cup \{\perp\}.$$

Intuizione

Gli oggetti “infiniti” non sono punti ma *blocchi intervallari unitari* $\infty_n = [n, n+1)$. La parte “infinita” del dominio è generata *solo* combinando questi blocchi *consecutivi* in unioni finite. Il simbolo $\perp$ è distinto: non è un intervallo né un'unione di intervalli.

Attenzione

In questo capitolo non incorporiamo ancora l'interazione completa con i numeri reali arbitrari. Limitiamo le operazioni al sotto-dominio $\mathbb{A}$ per fissare un *nucleo assiomatico* stabile. L'estensione mista (reali + infiniti etichettati) sarà sviluppata nei capitoli successivi.

3.2 Assiomi di base

Definizione 3.2 (Divisione per zero, caso determinato/indeterminato). Per $n \in \mathbb{N}$ con $n \geq 1$ si **definisce**

$$\frac{n}{0}\infty_n, \quad \text{dove } \infty_n = [n, n+1).$$

Per il caso $0/0$ si **definisce** l'**indeterminato**

$$\frac{0}{0}\perp,$$

che non appartiene a $\mathcal{U}$.

Definizione 3.3 (Estensionalità). Su $\mathcal{U}$ l'uguaglianza è *uguaglianza insiemistica*: per $U, V \in \mathcal{U}$,

$$U = V \iff (\forall x \in \mathbb{R}) (x \in U \iff x \in V).$$

Inoltre $\perp \neq U$ per ogni $U \in \mathcal{U}$.

Definizione 3.4 (Partizione e distinzione degli infiniti etichettati). Vale la **partizione**:

$$\bigcup_{n=1}^{\infty} \infty_n = [1, \infty).$$

Distinzione: per $m \neq n$ si ha $\infty_m \neq \infty_n$ (gli intervalli sono disgiunti).

3.3 Operazioni primitive su $\mathbb{A}$

Introduciamo tre operazioni primitive, tutte a valori in $\mathbb{A}$.

Definizione 3.5 (Somma (Minkowski) $\oplus$). Per $U, V \in \mathcal{U}$ definiamo

$$U \oplus V = \{u + v : u \in U, v \in V\}.$$

Per $\perp$ poniamo la **propagazione forte**:

$$U \oplus \perp = \perp \oplus U = \perp \quad \forall U \in \mathbb{A}.$$

Definizione 3.6 (Prodotto puntuale $\odot$). Per $U, V \in \mathcal{U}$ definiamo

$$U \odot V = \{u \cdot v : u \in U, v \in V\}.$$

Propagazione di $\perp$:

$$U \odot \perp = \perp \odot U = \perp \quad \forall U \in \mathbb{A}.$$

Definizione 3.7 (Scalatura intera positiva). Per $k \in \mathbb{N}$, $k \geq 1$ e $U \in \mathcal{U}$ definiamo

$$k \cdot U = \{k u : u \in U\}.$$

Propagazione di $\perp$:

$$k \cdot \perp = \perp \quad \text{per ogni } k \geq 1.$$

Attenzione

La propagazione “forte” di $\perp$ codifica che la presenza di 0/0 contamina l’operazione, mantenendo traccia dell’indeterminatezza. Varianti più deboli saranno discusse nelle appendici come alternative (con pro/contro).

3.4 Schemi di calcolo canonici

Gli schemi seguenti *calcolano esplicitamente* le tre operazioni su generatori ∞_n e su unioni consecutive, mostrando che il risultato appartiene ancora a $\mathcal{U}$ (chiusura).

Teorema 3.1 (Somma su generatori). Per $m, n \in \mathbb{N}$, $m, n \geq 1$,

$$\infty_m \oplus \infty_n = [m + n, m + n + 2) = \infty_{m+n} \cup \infty_{m+n+1}.$$

Teorema 3.2 (Scalatura di un generatore). Per $k \in \mathbb{N}$, $k \geq 1$, e $n \geq 1$,

$$k \cdot \infty_n = [kn, kn+k) = \bigcup_{i=0}^{k-1} \infty_{kn+i}.$$

Teorema 3.3 (Prodotto su generatori). Per $m, n \in \mathbb{N}$, $m, n \geq 1$,

$$\infty_m \odot \infty_n = [mn, (m+1)(n+1)) = \bigcup_{i=0}^{m+n} \infty_{mn+i}.$$

Teorema 3.4 (Estensione per unioni consecutive). Siano $U = [a, b+1) = \bigcup_{k=a}^b \infty_k$ e $V = [c, d+1) = \bigcup_{h=c}^d \infty_h$ con $1 \leq a \leq b$, $1 \leq c \leq d$. Allora:

$$U \oplus V = [a+c, b+d+2), \quad k \cdot U = [ka, kb+k), \quad U \odot V = [ac, (b+1)(d+1)).$$

In particolare $U \oplus V$, $k \cdot U$, $U \odot V \in \mathcal{U}$.

Esempio 3.1. $\infty_1 \oplus \infty_2 = [1, 2) \oplus [2, 3) = [3, 5) = \infty_3 \cup \infty_4$.

$$2 \cdot \infty_3 = [6, 8) = \infty_6 \cup \infty_7.$$

$$\infty_2 \odot \infty_3 = [6, 12) = \infty_6 \cup \dots \cup \infty_{11}.$$

Intuizione

Le tre regole “allargano” le etichette come ci si aspetta dall’aritmetica sugli intervalli: sommare due blocchi fa avanzare di *due* unità la lunghezza; scalare per k ripete k blocchi consecutivi; moltiplicare crea un blocco lungo $m+n+1$ che parte da mn .

3.5 Forma normale e semplificazione

Definizione 3.8 (Forma normale (FN)). Ogni $U \in \mathcal{U}$ ammette una **representazione unica** come intervallo consecutivo

$$\text{FN}(U) \equiv [p, q+1) = \bigcup_{k=p}^q \infty_k$$

con $1 \leq p \leq q$, detta *forma normale* di U .

Teorema 3.5 (Unicità della forma normale). *Se $[p, q + 1] = [p', q' + 1]$ con $1 \leq p \leq q$ e $1 \leq p' \leq q'$, allora $p = p'$ e $q = q'$.*

Definizione 3.9 (Semplificazione). Per espressioni costruite con $\oplus$, $\odot$, $\cdot$ su elementi di $\mathcal{U}$, la **forma normale** del risultato si ottiene applicando gli schemi della Sez. ?? seguiti da un'unica fusione in $[p, q + 1]$. Indichiamo tale operatore con $\text{FN}(\cdot)$.

Esempio 3.2. $\text{FN}((\infty_1 \oplus \infty_1) \oplus \infty_2) = \text{FN}([2, 4] \oplus [2, 3]) = \text{FN}([4, 7]) = [4, 7]$.

3.6 Indeterminato: regole di propagazione

Definizione 3.10 (Propagazione forte di $\perp$). Per ogni $U \in \mathbb{A}$ e per ogni $k \geq 1$,

$$U \oplus \perp = \perp \oplus U = \perp, \quad U \odot \perp = \perp \odot U = \perp, \quad k \cdot \perp = \perp.$$

Intuizione

Se un sotto-calcolo contiene $0/0$, l'intero risultato è marcato come indeterminato. Questa scelta *conservativa* evita di introdurre informazione fittizia.

Attenzione

In applicazioni specifiche si possono adottare politiche più fini (es. “isolamento locale” dell'indeterminato). Tali varianti sono opzionali e discusse in appendice, ma *non* fanno parte del nucleo assiomatico.

3.7 Chiusura e coerenza

Teorema 3.6 (Chiusura). *Per $U, V \in \mathcal{U}$ e $k \geq 1$ si ha $U \oplus V \in \mathcal{U}$, $U \odot V \in \mathcal{U}$ e $k \cdot U \in \mathcal{U}$.*

Teorema 3.7 (Coerenza con la partizione). *Ogni risultato $\in \mathcal{U}$ ammette una forma normale $[p, q+1)$ che è unione disgiunta di blocchi unitari $\infty_p, \dots, \infty_q$ della partizione $\bigcup_{n \geq 1} \infty_n = [1, \infty)$.*

Riassunto

Cosa abbiamo fissato in questo capitolo:

1. Gli oggetti “infiniti” sono blocchi $\infty_n = [n, n+1)$ e le loro unioni consecutive finite $\mathcal{U}$; $0/0 = \perp$.
2. Le operazioni primitive sono insiemistiche: Minkowski ($\oplus$), prodotto puntuale ($\odot$), scalatura intera ($k \cdot$).
3. Schemi chiusi e calcolabili: su generatori e su unioni consecutive, con forma normale unica $[p, q+1)$.
4. $\perp$ propaga in modo forte in ogni operazione del nucleo assiomatico.

Chapter 4

Regole Operative sugli Infiniti Etichettati

Riassunto

In sintesi: in questo capitolo fissiamo le *regole operative formali* per somma ($\oplus$), prodotto ($\odot$) e scalatura intera ($k \cdot$) sul dominio esteso $\mathbb{A} = \mathcal{U} \cup \{\perp\}$, dove $\mathcal{U} = \{[a, b+1) : 1 \leq a \leq b, a, b \in \mathbb{N}\}$ è l'insieme delle unioni finite *consecutive* di blocchi unitari $\infty_n = [n, n+1)$. Mostriamo formule chiuse, proprietà algebriche (commutatività, associatività, monotonia), un *algoritmo di forma normale* e tabelle operative per un calcolo rapido e coerente.

4.1 Richiamo e notazione

- $\infty_n = [n, n+1)$ per $n \in \mathbb{N}$, $n \geq 1$.
- Ogni $U \in \mathcal{U}$ è un unico intervallo consecutivo $U = [a, b+1) = \bigcup_{k=a}^b \infty_k$ con $1 \leq a \leq b$.
- $\perp$ è l'indeterminato (caso 0/0), distinto da ogni $U \in \mathcal{U}$.

4.2 Definizioni operative formali

Definizione 4.1 (Somma di Minkowski $\oplus$). Per $U, V \in \mathcal{U}$,

$$U \oplus V := \{u + v : u \in U, v \in V\} \in \mathcal{U}.$$

Propagazione di $\perp$:

$$U \oplus \perp = \perp \oplus U = \perp, \quad \forall U \in \mathbb{A}.$$

Definizione 4.2 (Prodotto puntuale $\odot$). Per $U, V \in \mathcal{U}$,

$$U \odot V := \{u \cdot v : u \in U, v \in V\} \in \mathcal{U}.$$

Propagazione di $\perp$:

$$U \odot \perp = \perp \odot U = \perp, \quad \forall U \in \mathbb{A}.$$

Definizione 4.3 (Scalatura intera positiva). Per $k \in \mathbb{N}$, $k \geq 1$, e $U \in \mathcal{U}$,

$$k \cdot U := \{k u : u \in U\} \in \mathcal{U}, \quad \text{e} \quad k \cdot \perp = \perp.$$

Intuizione

Pensiamo agli ∞_n come *blocchi di lunghezza 1* posizionati su $[1, \infty)$. Sommarli ($\oplus$) li “trasla e fonde” in un blocco più lungo; moltiplicarli ($\odot$) li “ripete”; moltiplicarli ($\cdot$) li *espande* secondo gli estremi prodotti.

4.3 Formule chiuse su generatori e su intervalli consecutivi

Teorema 4.1 (Somma su generatori). Per $m, n \geq 1$:

$$\infty_m \oplus \infty_n = [m, m+1) \oplus [n, n+1) = [m+n, m+n+2) = \infty_{m+n} \cup \infty_{m+n+1}.$$

Proof. Estremi inferiori e superiori si sommano: il minimo è $m+n$ (ottenuto con $m+n$), il massimo aperto è $(m+1) + (n+1) = m+n+2$. $\square$

Teorema 4.2 (Scalatura su generatori). Per $k \geq 1$ e $n \geq 1$:

$$k \cdot \infty_n = [kn, kn+k) = \bigcup_{i=0}^{k-1} \infty_{kn+i}.$$

Teorema 4.3 (Prodotto su generatori). Per $m, n \geq 1$:

$$\infty_m \odot \infty_n = [mn, (m+1)(n+1)) = \bigcup_{i=0}^{m+n} \infty_{mn+i}.$$

Teorema 4.4 (Formule su intervalli consecutivi). Siano $U = [a, b+1)$ e $V = [c, d+1)$ con $1 \leq a \leq b$ e $1 \leq c \leq d$. Allora:

$$U \oplus V = [a+c, b+d+2), \quad k \cdot U = [ka, kb+k), \quad U \odot V = [ac, (b+1)(d+1)).$$

Proof. Diretta dagli estremi min/max delle combinazioni possibili, essendo tutti i valori ≥ 1 . $\square$

Esempio 4.1. $\infty_1 \oplus \infty_2 = [1, 2) \oplus [2, 3) = [3, 5) = \infty_3 \cup \infty_4$. $2 \cdot \infty_3 = [6, 8) = \infty_6 \cup \infty_7$. $\infty_2 \odot \infty_3 = [6, 12) = \infty_6 \cup \dots \cup \infty_{11}$.

4.4 Proprietà algebriche

Teorema 4.5 (Commutatività e associatività). Per $U, V, W \in \mathcal{U}$ e $k \geq 1$ valgono:

$$\begin{aligned} U \oplus V &= V \oplus U, & (U \oplus V) \oplus W &= U \oplus (V \oplus W), \\ U \odot V &= V \odot U, & (U \odot V) \odot W &= U \odot (V \odot W), \\ k \cdot (U \oplus V) &= (k \cdot U) \oplus (k \cdot V), & k \cdot (U \odot V) &= (k \cdot U) \odot V = U \odot (k \cdot V). \end{aligned}$$

Proof. Seguono dalla commutatività/associatività di $+$ e $\cdot$ su $\mathbb{R}_{\geq 1}$ e dalla definizione insiemistica delle operazioni. Per la scalatura, $k(u+v) = ku + kv$ e $k(uv) = (ku)v = u(kv)$. $\square$

Teorema 4.6 (Monotonia). Se $U \subseteq U'$ e $V \subseteq V'$, allora:

$$U \oplus V \subseteq U' \oplus V', \quad U \odot V \subseteq U' \odot V', \quad k \cdot U \subseteq k \cdot U'.$$

Proof. Immediata dalle definizioni insiemistiche. $\square$

Attenzione

Attenzione alla (non) distributività mista. In generale $(U \oplus V) \odot W \subseteq (U \odot W) \oplus (V \odot W)$, ma non vale l'uguaglianza: il membro destro permette W “diversi” per i due termini U, V ; il sinistro impone lo *stesso* W .

Esempio 4.2 (Controesempio alla distributività mista). Sia $U = V = W = \infty_1 = [1, 2)$. Allora $(U \oplus V) \odot W = [2, 4) \odot [1, 2) = [2, 8)$. Invece $(U \odot W) \oplus (V \odot W) = [1, 4) \oplus [1, 4) = [2, 8)$.

Qui c'è uguaglianza per simmetria, ma scegliendo $U = [1, 2), V = [3, 4), W = [1, 1.1)$ (esempio continuo), si ottiene solo l'inclusione stretta. Nel nostro dominio (blocchi interi) l'inclusione è la regola generale.

Intuizione

$\oplus$ e $\odot$ sono “ben allineate” con l'aritmetica degli estremi: per $[a, b)$ e $[c, d)$, $[a, b) \oplus [c, d) = [a + c, b + d)$ e $[a, b) \odot [c, d) = [ac, bd)$ se $a, c \geq 1$. Questo spiega commutatività, associatività e monotonia.

4.5 Forma normale e algoritmo di calcolo

Definizione 4.4 (Forma normale (FN)). Ogni $U \in \mathcal{U}$ si scrive $\text{FN}(U) = [p, q+1)$ con $1 \leq p \leq q$ unici (una sola “barra” di blocchi unitari consecutivi).

Teorema 4.7 (Calcolo diretto in forma normale). Per $U = [a, b+1)$ e $V = [c, d+1)$:

$$\text{FN}(U \oplus V) = [a+c, b+d+2), \quad \text{FN}(k \cdot U) = [ka, kb+k), \quad \text{FN}(U \odot V) = [ac, (b+1)(d+1)).$$

Proof. Coincide con il Teo. ?? perché $\oplus, \odot, k \cdot$ preservano la consecutività. $\square$

Definizione 4.5 (Algoritmo FN (procedurale)). Dato un'espressione costruita con $\oplus, \odot, k \cdot$ su elementi in $\mathcal{U}$, per ottenere FN:

1. **Riscrivi** ogni simbolo ∞_n come $[n, n+1)$.

2. **Riduci bottom-up** usando le formule:

$$[a, b) \oplus [c, d) = [a+c, b+d), \quad k \cdot [a, b) = [ka, kb), \quad [a, b) \odot [c, d) = [ac, bd)$$

(valide per $a, b, c, d \geq 1$).

3. **Normalizza** il risultato in forma $[p, q+1)$ con $p, q \in \mathbb{N}$, $1 \leq p \leq q$.

Esempio 4.3 (FN passo-passo). $\text{FN}((2 \cdot \infty_2) \oplus (\infty_1 \odot \infty_3))$.

$$2 \cdot \infty_2 = [4, 6), \quad \infty_1 \odot \infty_3 = [3, 8), \quad [4, 6) \oplus [3, 8) = [7, 14) \equiv \bigcup_{k=7}^{13} \infty_k.$$

4.6 Tabelle operative

Somma $\oplus$ su generatori ($1 \leq m, n \leq 5$)

$\oplus$	∞_1	∞_2	∞_3	∞_4	∞_5
∞_1	[2, 4)	[3, 5)	[4, 6)	[5, 7)	[6, 8)
∞_2	[3, 5)	[4, 6)	[5, 7)	[6, 8)	[7, 9)
∞_3	[4, 6)	[5, 7)	[6, 8)	[7, 9)	[8, 10)
∞_4	[5, 7)	[6, 8)	[7, 9)	[8, 10)	[9, 11)
∞_5	[6, 8)	[7, 9)	[8, 10)	[9, 11)	[10, 12)

Prodotto $\odot$ su generatori ($1 \leq m, n \leq 5$)

$\odot$	∞_1	∞_2	∞_3	∞_4	∞_5
∞_1	[1, 4)	[2, 6)	[3, 8)	[4, 10)	[5, 12)
∞_2	[2, 6)	[4, 9)	[6, 12)	[8, 15)	[10, 18)
∞_3	[3, 8)	[6, 12)	[9, 16)	[12, 20)	[15, 24)
∞_4	[4, 10)	[8, 15)	[12, 20)	[16, 25)	[20, 30)
∞_5	[5, 12)	[10, 18)	[15, 24)	[20, 30)	[25, 36)

Scalatura $k \cdot \infty_n$ ($k \leq 5, n \leq 5$)

$k \backslash n$	1	2	3	4	5
1	[1, 2)	[2, 3)	[3, 4)	[4, 5)	[5, 6)
2	[2, 4)	[4, 6)	[6, 8)	[8, 10)	[10, 12)
3	[3, 6)	[6, 9)	[9, 12)	[12, 15)	[15, 18)
4	[4, 8)	[8, 12)	[12, 16)	[16, 20)	[20, 24)
5	[5, 10)	[10, 15)	[15, 20)	[20, 25)	[25, 30)

Intuizione

Le tabelle rendono immediato il calcolo: basta combinare gli estremi come da formule chiuse. Ogni risultato può essere, se desiderato, riscritto come unione $\infty_p \cup \dots \cup \infty_q$.

4.7 Confronto e misura di ampiezza

Definizione 4.6 (Inizio, fine, ampiezza). Per $U = [a, b+1)$ poniamo $\text{start}(U) = a$, $\text{end}(U) = b + 1$ e $\text{len}(U) = \text{end}(U) - \text{start}(U) = b - a + 1$, cioè il *numero di blocchi unitari* contenuti in U .

Definizione 4.7 (Ordine lessicografico). Definiamo $U \preceq V$ se $\text{start}(U) < \text{start}(V)$, oppure $\text{start}(U) = \text{start}(V)$ e $\text{len}(U) \leq \text{len}(V)$.

Proposizione 4.8 (Monotonia degli estremi). Se $U \preceq V$ allora, per ogni $W \in \mathcal{U}$ e $k \geq 1$,

$$\text{start}(U \oplus W) \leq \text{start}(V \oplus W), \quad \text{start}(U \odot W) \leq \text{start}(V \odot W), \quad \text{start}(k \cdot U) \leq \text{start}(k \cdot V).$$

Analoghe disuguaglianze valgono per $\text{end}(\cdot)$.

Proof. Diretta dalle formule chiuse su estremi. □

4.8 Linee guida operative

Riassunto

Per calcolare velocemente:

1. Riscrivi ogni ∞_n come $[n, n + 1)$.
2. Applica le *formule chiuse* agli estremi (Somma: somma degli estremi; Prodotto: prodotti degli estremi; Scalatura: moltiplica gli estremi per k).
3. Riporta il risultato in FN come $[p, q + 1)$ e, se utile, come unione $\infty_p \cup \dots \cup \infty_q$.
4. Se compare $\perp$ in qualunque sotto-espressione, il risultato è $\perp$.

Attenzione

Checklist di coerenza:

- Tutti gli estremi devono rimanere ≥ 1 (garantito dal dominio).
- $\oplus, \odot, k \cdot$ preservano la consecutività: il risultato è sempre un singolo intervallo $[p, q + 1)$.
- Non confondere la distributività *mista* (in generale solo inclusione $\subseteq$).

4.9 Esercizi consigliati

1. Calcola in FN: $(\infty_2 \oplus \infty_4) \odot \infty_3$.
2. Dimostra l'associatività di $\oplus$ usando unicamente la definizione insiemistica.
3. Mostra che $k \cdot (\infty_m \oplus \infty_n) = (k \cdot \infty_m) \oplus (k \cdot \infty_n)$ con le formule degli estremi.
4. Trova un esempio concreto in cui $(U \oplus V) \odot W \subsetneq (U \odot W) \oplus (V \odot W)$.
5. Per $U = [a, b+1)$ e $V = [c, d+1)$, verifica $\text{len}(U \oplus V) = \text{len}(U) + \text{len}(V) + 1$ e $\text{len}(U \odot V) = \text{len}(U) + \text{len}(V) - 1$.

Intuizione

Le lunghezze si combinano in modo lineare: la somma “aggiunge” una unità, il prodotto “somma” le lunghezze e poi sottrae una, coerentemente con gli estremi: $[a, b] \oplus [c, d] = [a + c, b + d]$, $[a, b] \odot [c, d] = [ac, bd]$.

Chapter 5

Teoremi principali

Riassunto

In sintesi: qui raccogliamo i risultati fondamentali che garantiscono la correttezza, la solidità e l'effettività del nucleo operativo di AIE2 (dominio $\mathbb{A} = \mathcal{U} \cup \{\perp\}$). I teoremi principali sono:

- esistenza ed unicità della forma normale (FN) per ogni espressione del linguaggio del nucleo;
- corrispondenza operatore $\leftrightarrow$ operazioni sugli estremi (start/end) — che rende il calcolo effettivo;
- proprietà algebriche globali (chiusura, monotonia, commutatività, associatività);
- correttezza e terminazione dell'algoritmo di valutazione in forma normale;
- stime qualitative su crescita degli estremi e complessità aritmetica.

5.1 Notazione rapida e premesse

Ricordiamo la notazione usata in questo capitolo:

- $\infty_n = [n, n + 1)$ per $n \geq 1$.
- ogni elemento $U \in \mathcal{U}$ si rappresenta univocamente come $U = [a, b + 1)$ con $1 \leq a \leq b$ (forma normale).
- definiamo le funzioni $\text{start}, \text{end} : \mathcal{U} \rightarrow \mathbb{N}$ per $U = [a, b + 1)$ come $\text{start}(U) = a$, $\text{end}(U) = b + 1$.
- la lunghezza $\text{len}(U) = \text{end}(U) - \text{start}(U) = b - a + 1$ (numero di blocchi).

5.2 Corrispondenza operativa sugli estremi (mappa start/end)

Il primo insieme di risultati mostra che ciascuna operazione su intervalli si traduce in una semplice operazione aritmetica sugli *estremi* (start,end). Questo è il ponte fondamentale per rendere FN computabile ed efficace.

Teorema 5.1 (Corrispondenza degli estremi). *Siano $U = [a, b + 1)$ e $V = [c, d + 1)$ con $1 \leq a \leq b$, $1 \leq c \leq d$. Allora valgono:*

- (a) *Somma:* $U \oplus V = [a + c, b + d + 2),$
- (b) *Scalatura:* $k \cdot U = [ka, k(b + 1)), \quad k \in \mathbb{N}, k \geq 1,$
- (c) *Prodotto:* $U \odot V = [ac, (b + 1)(d + 1)).$

Equivalentemente, in termini di start/end:

$$\begin{aligned} \text{start}(U \oplus V) &= \text{start}(U) + \text{start}(V), & \text{end}(U \oplus V) &= \text{end}(U) + \text{end}(V), \\ \text{start}(k \cdot U) &= k \cdot \text{start}(U), & \text{end}(k \cdot U) &= k \cdot \text{end}(U), \\ \text{start}(U \odot V) &= \text{start}(U) \cdot \text{start}(V), & \text{end}(U \odot V) &= \text{end}(U) \cdot \text{end}(V). \end{aligned}$$

Proof. Le formule sono ottenute direttamente dalle definizioni di $\oplus, \odot, k \cdot$ come operazioni insiemistiche sugli intervalli:

- Se $u \in [a, b + 1)$ e $v \in [c, d + 1)$ allora $a + c \leq u + v < (b + 1) + (d + 1) = b + d + 2$, da cui (a).
- Se $u \in [a, b + 1)$ allora $ka \leq ku < k(b + 1)$, da cui (b).
- Se $u \in [a, b + 1)$ e $v \in [c, d + 1)$ allora $ac \leq uv < (b + 1)(d + 1)$, da cui (c).

Le uguaglianze sugli start/end seguono immediatamente. □

[Formule per le lunghezze] Per gli stessi U, V valgono:

$$\begin{aligned} \text{len}(U \oplus V) &= \text{len}(U) + \text{len}(V), & \text{len}(k \cdot U) &= k \cdot \text{len}(U), \\ \text{len}(U \odot V) &= \text{end}(U)\text{end}(V) - \text{start}(U)\text{start}(V). \end{aligned}$$

Proof. Ricordando $\text{len}(U) = \text{end}(U) - \text{start}(U)$ si ottengono le formule applicando il TEO. ?? alle definizioni. □

5.3 Rappresentazione unica e forma normale

Qui formalizziamo l'enunciato fondamentale: ogni espressione del linguaggio del nucleo (costruita con generatori ∞_n , $\perp$ e operatori $\oplus, \odot, k\cdot$) riduce in modo unico a un elemento di $\mathbb{A}$.

Definizione 5.1 (Linguaggio del nucleo). Denotiamo con $\mathcal{L}$ l'insieme delle espressioni generate da:

$$E ::= \infty_n \ (n \geq 1) \mid \perp \mid E \oplus E \mid E \odot E \mid k \cdot E \quad (k \in \mathbb{N}, k \geq 1).$$

Teorema 5.2 (Esistenza ed unicità della valutazione in FN). *Esiste una funzione computabile*

$$\text{Eval} : \mathcal{L} \rightarrow \mathbb{A}$$

tale che per ogni espressione $e \in \mathcal{L}$:

1. *se e contiene $\perp$ come sotto-espressione allora $\text{Eval}(e) = \perp$;*
2. *altrimenti $\text{Eval}(e) = [p, q + 1)$ per un unico $p, q \in \mathbb{N}$ (forma normale);*
3. *l'algoritmo di valutazione è ottenuto valutando ricorsivamente start, end con le regole del TEO. ??.*

In particolare la forma normale $\text{FN}(\text{Eval}(e))$ è unica.

Proof. Costruiamo Eval per ricorsione strutturale sull'albero sintattico di e :

- caso base: $\text{Eval}(\infty_n) = [n, n + 1)$ e $\text{Eval}(\perp) = \perp$;
- passo ricorsivo: se $e = e_1 \oplus e_2$ allora calcoliamo $U_i = \text{Eval}(e_i)$. Se uno dei U_i è $\perp$ allora $\text{Eval}(e) = \perp$; altrimenti $U_i = [a_i, b_i + 1)$ e si pone $\text{Eval}(e) = [a_1 + a_2, b_1 + b_2 + 2)$ (teorema ??). Analogamente per $\odot$ e $k\cdot$.

L'algoritmo termina perché ogni passo riduce la profondità dell'albero (terminazione per ricorsione strutturale). Unicità: la rappresentazione come $[p, q + 1)$ è determinata univocamente dagli interi $p = \text{start}$ e $q + 1 = \text{end}$. Dunque due valutazioni con gli stessi start, end definiscono lo stesso elemento. $\square$

Il fatto che la valutazione si riduca a calcoli su start/end è la chiave pratico/algoritmica che rende FN semplice da implementare (vedi Cap. ?? per la formalizzazione in Lean).

5.4 Proprietà algebriche globali (riepilogo e corollari)

Molte proprietà già discusse si ricavano ora in modo diretto dalla rappresentazione sugli estremi.

[Chiusura] $\mathbb{A}$ è chiuso rispetto a $\oplus, \odot, k \cdot$: se $U, V \in \mathbb{A}$ e $k \geq 1$ allora $U \oplus V, U \odot V, k \cdot U \in \mathbb{A}$ (con la convenzione di propagazione di $\perp$).

[Commutatività e associatività] Gli operatori $\oplus$ e $\odot$ sono commutativi e associativi su $\mathcal{U}$ (e si estendono ovviamente a $\mathbb{A}$ con la regola $\perp$ che propaga).

Proof. Dirette dalle identità su start/end: addizione e moltiplicazione degli interi sono commutative e associative e le formule del TEO. ?? rispettano questa struttura. $\square$

[Monotonia] Se $U \subseteq U'$ e $V \subseteq V'$ allora $U \oplus V \subseteq U' \oplus V', U \odot V \subseteq U' \odot V', k \cdot U \subseteq k \cdot U'$.

Proof. Segue dal fatto che le funzioni start/end sono monotone (start maggiore implica start prodotto maggiore, ...). $\square$

Attenzione

Non esiste, in generale, una distributività inversa perfetta (ad es. $(U \oplus V) \odot W$ può essere strettamente incluso in $(U \odot W) \oplus (V \odot W)$); si conservi la regola di inclusione e si verifichino i casi particolari quando la uguaglianza è richiesta.

5.5 Terminazione e complessità della valutazione in FN

Terminazione L'algoritmo Eval definito nella prova del TEO. ?? termina per ricorsione strutturale sull'albero sintattico: ogni chiamata valuta sotto-problemi di dimensione strettamente minore.

Complessità (osservazioni pratiche). Sia n il numero totale di nodi (leaf + operatori) dell'espressione e . Sia B la massima lunghezza in bit (la *bit-size*) degli interi presenti come parametri di base (gli indici n dei generatori ∞_n o i coefficienti k). Allora:

- il numero di operazioni aritmetiche (addizioni e moltiplicazioni di interi) necessarie è $O(n)$ (ogni nodo genera un numero costante di operazioni sugli estremi);
- la bit-size degli interi prodotti (start, end intermedi) cresce *al più* additivamente lungo i percorsi di moltiplicazione: la log-magnitudine dell'intero risultante è al più la somma delle log-magnitudine degli operandi (log prodotto = somma log). Quindi la bit-size di qualsiasi estremità intermedia è $O(n \cdot B)$.
- la complessità totale (con algoritmo di moltiplicazione $Mul(b)$ che moltiplica due interi di bit-length b in tempo $Mul(b)$) è quindi $O(n \cdot Mul(nB))$. Con l'algoritmo naïf di moltiplicazione $Mul(b) = O(b^2)$ si ottiene $O(n \cdot (nB)^2)$; con algoritmi più veloci la complessità diminuisce proporzionalmente.

Queste stime sono conservative ma utili per valutare fattibilità pratica: la valutazione è *effettiva* e, per espressioni di dimensione ragionevole con parametri di bit-size moderata, computabile in tempi polinomiali.

5.6 Teorema di rappresentazione (omomorfismo start/end)

La mappa che associa ad ogni espressione e la coppia di interi $(\text{start}(\text{Eval}(e)), \text{end}(\text{Eval}(e)))$ è un omomorfismo tra l'algebra sintattica generata da $(\oplus, \odot, k \cdot)$ e l'algebra dei coppie di interi con operazioni component-wise indicate dalle formule del TEO. ??.

Teorema 5.3 (Omomorfismo start/end). *Sia H la funzione che ad ogni espressione $e \in \mathcal{L}$ associa la coppia di interi*

$$H(e) := (\text{start}(\text{Eval}(e)), \text{end}(\text{Eval}(e))).$$

Allora H soddisfa, per tutte le espressioni e_1, e_2 :

$$H(e_1 \oplus e_2) = H(e_1) \oplus_{\mathbb{N}^2} H(e_2), \quad H(e_1 \odot e_2) = H(e_1) \odot_{\mathbb{N}^2} H(e_2),$$

$$H(k \cdot e) = k \cdot_{\mathbb{N}^2} H(e),$$

dove le operazioni sul lato destro sono definite componente per componente come:

$$(a_1, a_2) \oplus_{\mathbb{N}^2} (b_1, b_2) := (a_1 + b_1, a_2 + b_2),$$

$$(a_1, a_2) \odot_{\mathbb{N}^2} (b_1, b_2) := (a_1 b_1, a_2 b_2),$$

$$k \cdot_{\mathbb{N}^2} (a_1, a_2) := (ka_1, ka_2).$$

Proof. Segue in maniera diretta dalle formule del TEO. ?? e dalla definizione di H . $\square$

[Unicità tramite omomorfismo] Poiché un intervallo $[p, q)$ è determinato univocamente dalla coppia (p, q) , il teorema precedente implica che se due espressioni hanno la stessa immagine tramite H allora valutano allo stesso elemento di $\mathcal{U}$. Questo fornisce un criterio pratico per il testing di uguaglianza simbolica nel nucleo: uguaglianza di coppie start/end.

5.7 Esempi applicativi e controesempi illustrativi

Esempio 5.1 (Esempio diretto). Calcoliamo in FN l'espressione

$$e = (2 \cdot \infty_2) \oplus (\infty_1 \odot \infty_3).$$

Valutiamo gli estremi:

$$H(2 \cdot \infty_2) = (4, 6), \quad H(\infty_1 \odot \infty_3) = (3, 8).$$

Quindi

$$H(e) = (4 + 3, 6 + 8) = (7, 14),$$

e pertanto $\text{Eval}(e) = [7, 14) = \bigcup_{k=7}^{13} \infty_k$.

Esempio 5.2 (Controesempio alla distributività (conclusione)). Sia $U = [1, 2)$, $V = [3, 4)$, $W = [1, 1 + \varepsilon)$ con $0 < \varepsilon < 1$. Nel dominio reale continuo (fuori da $\mathcal{U}$) è possibile mostrare che $(U \oplus V) \odot W$ può essere strettamente incluso in $(U \odot W) \oplus (V \odot W)$. Nel nostro nucleo discreto (blocchi interi) si possono costruire esempi analoghi se si permettono estremi non interi (o scalature non compatte): la regola sicura e generale è l'inclusione $(U \oplus V) \odot W \subseteq (U \odot W) \oplus (V \odot W)$.

5.8 Bozza di formalizzazione in Lean (schema)

Riassunto

Obiettivo: suggerire uno schema di rappresentazione del nucleo in Lean (idea, non codice completo).

```
-- tipi base
structure Interval := (start : ) (end_ : ) -- con 1 start < end_
inductive AIEValue
| interval : Interval → AIEValue
| indet    : AIEValue

-- funzioni ausiliarie
def start_of : AIEValue → option := ...
```

```

def end_of      : AIEValue → option  := ...

-- operatori definibili in termini di start/end (con propagazione di indet)
def add (x y : AIEValue) : AIEValue := ...
def mul (x y : AIEValue) : AIEValue := ...
def scale (k : ) (x : AIEValue) : AIEValue := ...

-- lemmi: correctness (start/end formulas), closure, associativity, commutativity
theorem add_start_end (x y : AIEValue) : ... := ...

```

Queste definizioni/lemmi corrispondono ai teoremi precedenti e sono immediatamente formalizzabili perché lavorano con operazioni aritmetiche su naturali (start,end).

5.9 Riepilogo e consegne operative

Riassunto

Cosa portare via da questo capitolo:

1. La valutazione di ogni espressione del nucleo si riduce al calcolo di due interi (start,end).
2. Le operazioni sono omomorfe rispetto alla mappa start/end: ciò rende la normale forma unica e la valutazione effettiva e deterministica.
3. L'algoritmo di valutazione termina e richiede un numero di operazioni aritmetiche lineare nella dimensione sintattica dell'espressione.
4. Le stime sulla crescita degli interi prodotti permettono di ragionare sulla complessità pratica; l'implementazione in un proof-assistant (Lean) è diretta.

Esercizi consigliati

1. Dimostrare formalmente (per induzione sulla struttura) il TEO. ??.
2. Implementare in pseudocode la funzione H (start,end) e analizzarne la complessità in dettaglio.
3. Dare un esempio di espressione che mostra crescita non banale della bit-size degli estremi e stimare la crescita in funzione della profondità dell'albero.
4. Formalizzare in Lean la struttura `Interval` e provare i lemmi di commutatività per $\oplus$ e $\odot$.

Chapter 6

Esempi numerici e tabelle operative (senza diagrammi)

Riassunto

In sintesi: in questo capitolo forniamo esempi numerici completi, calcoli passo–passo e tabelle operative estese per mostrare come usare concretamente le regole di AIE2 senza ricorrere a diagrammi. Ogni esempio termina con la *Forma Normale* (FN) del risultato e una breve interpretazione. Questo capitolo è pensato come riferimento rapido.

6.1 Metodo e convenzioni per i calcoli

Per chiarezza operativa adottiamo le seguenti convenzioni in tutti gli esempi:

- Indichiamo ogni elemento di $\mathcal{U}$ nella forma compatta $[s, e)$ con $s = \text{start}$, $e = \text{end}$. Ricordiamo che nella nostra notazione fondamentale un generatore $\infty_n = [n, n + 1)$ ha quindi $\text{start} = n$ e $\text{end} = n + 1$.

- Per la somma, scalatura e prodotto usiamo le formule sugli estremi (Teo. ??):

$$[a, b) \oplus [c, d) = [a+c, b+d), \quad k \cdot [a, b) = [ka, kb), \quad [a, b) \odot [c, d) = [ac, bd).$$

(Qui b, d sono gli *end* come definiti sopra.)

- Dopo ogni operazione riportiamo il risultato in *Forma Normale* (FN), cioè come $[p, q)$ e — se utile — come unione di generatori $\infty_p \cup \dots \cup \infty_{q-1}$.
- Se compare $\perp$ in una sotto-espressione, il risultato dell'intera espressione del nucleo è $\perp$.

6.2 Tabelle operative estese (esempi numerici)

Somma $\oplus$ — tabella estesa ($1 \leq m, n \leq 8$)

$\oplus$	∞_1	∞_2	∞_3	∞_4	∞_5	∞_6	∞_7	∞_8
∞_1	[2, 4)	[3, 5)	[4, 6)	[5, 7)	[6, 8)	[7, 9)	[8, 10)	[9, 11)
∞_2	[3, 5)	[4, 6)	[5, 7)	[6, 8)	[7, 9)	[8, 10)	[9, 11)	[10, 12)
∞_3	[4, 6)	[5, 7)	[6, 8)	[7, 9)	[8, 10)	[9, 11)	[10, 12)	[11, 13)
∞_4	[5, 7)	[6, 8)	[7, 9)	[8, 10)	[9, 11)	[10, 12)	[11, 13)	[12, 14)
∞_5	[6, 8)	[7, 9)	[8, 10)	[9, 11)	[10, 12)	[11, 13)	[12, 14)	[13, 15)
∞_6	[7, 9)	[8, 10)	[9, 11)	[10, 12)	[11, 13)	[12, 14)	[13, 15)	[14, 16)
∞_7	[8, 10)	[9, 11)	[10, 12)	[11, 13)	[12, 14)	[13, 15)	[14, 16)	[15, 17)
∞_8	[9, 11)	[10, 12)	[11, 13)	[12, 14)	[13, 15)	[14, 16)	[15, 17)	[16, 18)

Prodotto $\odot$ — tabella estesa ($1 \leq m, n \leq 8$)

$\odot$	∞_1	∞_2	∞_3	∞_4	∞_5	∞_6	∞_7	∞_8
∞_1	[1, 4)	[2, 6)	[3, 8)	[4, 10)	[5, 12)	[6, 14)	[7, 16)	[8, 18)
∞_2	[2, 6)	[4, 9)	[6, 12)	[8, 15)	[10, 18)	[12, 21)	[14, 24)	[16, 27)
∞_3	[3, 8)	[6, 12)	[9, 16)	[12, 20)	[15, 24)	[18, 28)	[21, 32)	[24, 36)
∞_4	[4, 10)	[8, 15)	[12, 20)	[16, 25)	[20, 30)	[24, 36)	[28, 42)	[32, 50)
∞_5	[5, 12)	[10, 18)	[15, 24)	[20, 30)	[25, 36)	[30, 42)	[35, 54)	[40, 60)
∞_6	[6, 14)	[12, 21)	[18, 28)	[24, 36)	[30, 42)	[36, 49)	[42, 56)	[48, 63)
∞_7	[7, 16)	[14, 24)	[21, 32)	[28, 42)	[35, 54)	[42, 56)	[49, 64)	[56, 72)
∞_8	[8, 18)	[16, 27)	[24, 36)	[32, 50)	[40, 60)	[48, 63)	[56, 72)	[64, 81)

Scalatura $k \cdot \infty_n$ ($1 \leq k \leq 6, 1 \leq n \leq 8$)

$k \backslash n$	1	2	3	4	5	6	7	8
1	[1, 2)	[2, 3)	[3, 4)	[4, 5)	[5, 6)	[6, 7)	[7, 8)	[8, 9)
2	[2, 4)	[4, 6)	[6, 8)	[8, 10)	[10, 12)	[12, 14)	[14, 16)	[16, 18)
3	[3, 6)	[6, 9)	[9, 12)	[12, 15)	[15, 18)	[18, 21)	[21, 24)	[24, 27)
4	[4, 8)	[8, 12)	[12, 16)	[16, 20)	[20, 24)	[24, 28)	[28, 32)	[32, 36)
5	[5, 10)	[10, 15)	[15, 20)	[20, 25)	[25, 30)	[30, 35)	[35, 40)	[40, 45)
6	[6, 12)	[12, 18)	[18, 24)	[24, 30)	[30, 36)	[36, 42)	[42, 48)	[48, 54)

6.3 Esempi calcolati passo–passo

Esempio 1 — Somma semplice

Calcolare $\infty_1 \oplus \infty_1$.

Calcolo:

$$\infty_1 = [1, 2), \quad \infty_1 \oplus \infty_1 = [1 + 1, 2 + 2) = [2, 4).$$

Forma Normale: $[2, 4) = \infty_2 \cup \infty_3$.

Interpretazione: la somma di due "infinito_1" produce due blocchi successivi, etichettati 2 e 3.

Esempio 2 — Scalatura e somma

Calcolare $(2 \cdot \infty_3) \oplus \infty_1$.

Passo 1: scalatura

$$2 \cdot \infty_3 = [6, 8).$$

Passo 2: somma con $\infty_1 = [1, 2)$

$$[6, 8) \oplus [1, 2) = [6 + 1, 8 + 2) = [7, 10).$$

Forma Normale: $[7, 10) = \infty_7 \cup \infty_8 \cup \infty_9$.

Interpretazione: la scalatura ha "spostato" l'intervallo a partire da 6 e la somma lo ha avanzato ulteriormente.

Esempio 3 — Espressione annidata con prodotto

Calcolare $e = (\infty_1 \oplus (2 \cdot \infty_3)) \odot \infty_2$.

Passo 1: calcolare $2 \cdot \infty_3 = [6, 8)$.

Passo 2: somma interna

$$\infty_1 \oplus [6, 8) = [1 + 6, 2 + 8) = [7, 10).$$

Passo 3: prodotto con $\infty_2 = [2, 3)$:

$$[7, 10) \odot [2, 3) = [7 \cdot 2, 10 \cdot 3) = [14, 30).$$

Forma Normale: $[14, 30) = \infty_{14} \cup \dots \cup \infty_{29}$.

Interpretazione: il prodotto espande l'intervallo in modo consistente con gli estremi: start moltiplicato per start, end per end.

Esempio 4 — Propagazione di $\perp$

Calcolare $f = (\infty_2 \odot \perp) \oplus \infty_1$.

Regola: la presenza di $\perp$ in una sotto-espressione fa propagare $\perp$.

Valutazione:

$$\infty_2 \odot \perp = \perp, \quad \text{quindi} \quad f = \perp \oplus \infty_1 = \perp.$$

Forma Normale: $\perp$.

Interpretazione: l'indeterminatezza di un fattore rende indeterminata l'intera espressione (scelta conservativa del nucleo).

Esempio 5 — Verifica di uguaglianza tramite la mappa H

Mostrare che $(\infty_1 \oplus \infty_3) \oplus \infty_2$ e $\infty_1 \oplus (\infty_3 \oplus \infty_2)$ valutano allo stesso elemento.

Calcolo sinistro:

$$\infty_1 \oplus \infty_3 = [4, 6), \quad [4, 6) \oplus \infty_2 = [4 + 2, 6 + 3) = [6, 9).$$

Quindi $H(\text{sinistro}) = (6, 9)$.

Calcolo destro:

$$\infty_3 \oplus \infty_2 = [5, 7), \quad \infty_1 \oplus [5, 7) = [1 + 5, 2 + 7) = [6, 9).$$

Quindi $H(\text{destro}) = (6, 9)$.

Conclusione: le coppie start/end coincidono, dunque le due espressioni sono uguali per il nucleo (associatività confermata).

6.4 Strategie di compressione o rappresentante per unione di intervalli

Nei casi pratici può essere utile associare ad ogni intervallo $U = [p, q)$ un *rappresentante unico* (singolo indice) per semplificare confronti, display, o politiche di tie-breaking. Di seguito alcune strategie standard:

1. **Rappresentante di start (minimale):** $r_{\text{start}}(U) := p$. Questo privilegia l'origine dell'intervallo.
2. **Rappresentante centrale (centroide discreto):** $r_{\text{mid}}(U) := \left\lfloor \frac{p + (q - 1)}{2} \right\rfloor$.
Restituisce l'indice del blocco centrale (utile per visualizzazione compatta).
3. **Rappresentante di lunghezza (massimo):** se si vuole indicare la "taglia" si può usare $r_{\text{len}}(U) := \text{len}(U) = q - p$ (numero di blocchi).
4. **Compressione parametrica:** fissato un parametro $h \geq 1$ si può mappare U in $\tilde{r}(U) = \left\lfloor \frac{p}{h} \right\rfloor$ per raggruppare intervalli in bucket di ampiezza h .

Esempio 6.1. Per $U = [7, 14)$ (esempio Esempio 3), abbiamo:

$$r_{\text{start}}(U) = 7, \quad r_{\text{mid}}(U) = \left\lfloor \frac{7 + 13}{2} \right\rfloor = 10, \quad r_{\text{len}}(U) = 7.$$

Intuizione

La scelta del rappresentante dipende dallo scopo: per calcoli simbolici preferiamo r_{start} (deterministico); per visualizzazione o clustering è spesso più utile r_{mid} .

6.5 Esempi applicativi rapidi (interpretazioni pratiche)

Uso come segnalatore di contributi divergenti

Immaginiamo due contributi di un modello fisico che, senza regolarizzazione, divergerebbero separatamente. In AIE2 possono essere rappresentati come ∞_{n_1} e ∞_{n_2} . Le operazioni tra contributi (somma, prodotto) producono un intervallo che indica il supporto numerico possibile del risultato divergente — questo facilita confronti qualitativi e politiche di regolarizzazione mirate (scegliere la regola R più adatta in funzione della posizione start/end).

Uso in overflow semantico

In informatica simbolica, se una funzione produce ∞_m come risultato di overflow, la propagazione di questa informazione consente alle funzioni chiamanti di prendere decisioni strutturate (es. usare la compressione o eseguire fallback), invece di ricevere un semplice NaN non informativo.

6.6 Esercizi con soluzioni guidate

1. Calcolare in FN: $(\infty_2 \oplus \infty_4) \odot (\infty_1 \oplus \infty_3)$.

Soluzione guidata: $\infty_2 \oplus \infty_4 = [6, 8)$, $\infty_1 \oplus \infty_3 = [4, 6)$. Quindi prodotto $= [6 \cdot 4, 8 \cdot 6) = [24, 48)$.

2. Dimostrare che $k \cdot (\infty_m \oplus \infty_n) = (k \cdot \infty_m) \oplus (k \cdot \infty_n)$ per ogni $k \geq 1$.

Soluzione guidata: usare formule sugli estremi: start sinistro $= k(m + n) =$ start destro, analoga per end.

3. Trovare un esempio dove $(U \oplus V) \odot W \subsetneq (U \odot W) \oplus (V \odot W)$ nel senso di estremi (fornire i tre intervalli).

Suggerimento: scegliere $U = [1, 2)$, $V = [3, 4)$, $W = [1, 1.5)$ (nota: questo esempio usa estremi non interi per mostrare la differenza teorica;

nel nucleo discreto usare invece casi con scalature non compatte per ottenere l'inclusione stretta).

4. Per $e = (\infty_1 \odot \infty_4) \oplus (3 \cdot \infty_2)$ trovare $\text{FN}(e)$ e il rappresentante r_{mid} .

Soluzione guidata: $\infty_1 \odot \infty_4 = [4, 10)$, $3 \cdot \infty_2 = [6, 9)$. Somma $[10, 19)$. $\text{FN} = [10, 19)$. $r_{\text{mid}} = \lfloor (10 + 18)/2 \rfloor = 14$.

5. Mostrare con calcoli che la valutazione è indipendente dall'ordine di applicazione per $\oplus$ (associatività verificata numeric.)

Suggerimento: ripetere Esempio 5 con altri indici a piacere e confrontare coppie start/end.

6.7 Riferimenti rapidi alle regole usate

- Somma: $[a, b) \oplus [c, d) = [a + c, b + d)$.
- Scalatura: $k \cdot [a, b) = [ka, kb)$.
- Prodotto: $[a, b) \odot [c, d) = [ac, bd)$.
- Forma Normale: rappresentare il risultato come $[p, q)$ e, se si vuole, come $\bigcup_{k=p}^{q-1} \infty_k$.
- Propagazione di $\perp$: qualunque operazione contenente $\perp$ restituisce $\perp$.

Riassunto

Conclusione del capitolo: questo capitolo fornisce una libreria pratica di esempi numerici e tabelle operative per applicare AIE2 nelle manipolazioni simboliche del nucleo. Senza diagrammi abbiamo mostrato passo per passo come ottenere la Forma Normale e come interpretare i risultati operativi; le stesse tecniche sono immediatamente implementabili in codice.

Chapter 7

Confronto con altre teorie matematiche

Riassunto

In sintesi: questo capitolo mette a confronto AIE2 con teorie e strumenti consolidati che trattano infiniti, infinitesimi o regolarizzazione (Analisi Non Standard, sfera di Riemann / infinito proiettivo, metodi di regolarizzazione classici, logica modale, teoria di Galois). L'obiettivo non è ridurre una teoria all'altra ma:

- chiarire analogie e differenze formali e concettuali;
- valutare punti di forza/debolezza di AIE2 rispetto a ciascuna alternativa;
- fornire indicazioni pratiche su quando preferire AIE2 o una teoria concorrente.

7.1 Premessa

AIE2 è una *proposta operativa* che aggiunge etichette e una struttura intervallare agli oggetti "infiniti". Molte teorie esistenti affrontano problemi simili (gestione delle divergenze, infinitesimi, punti all'infinito), ma con obiettivi e strumenti spesso diversi (astrazione assiomatica, struttura algebrica, strumenti analitici). Nelle sezioni seguenti confrontiamo AIE2 con alcune di queste teorie su vari piani: semantico, algebrico, computazionale e applicativo.

7.2 Confronto con l'Analisi Non Standard (NSA)

Richiamo rapido di NSA

L'Analisi Non Standard (Robinson e successori) produce un campo esteso ${}^*\mathbb{R}$ contenente infinitesimi e infiniti. Ha una potente *transfer principle* che garantisce che ogni proprietà (definibile nella lingua dei reals) si trasferisca ai corrispondenti iperreali. La costruzione classica usa ultrafilter/ultraprodotti e richiede strumenti set-theoretici non costruttivi (asincronia con la computabilità immediata).

Analogie

- Entrambe le teorie introducono *entità non-standard* che permettono di parlare di grandezze "infinite" o "infinitesime" in modo formale.
- In certe interpretazioni concettuali si può vedere un parallelo: un elemento infinitamente grande di ${}^*\mathbb{R}$ è qualitativamente simile all'idea di una ∞_n che rappresenta una spinta verso l'infinito — entrambe codificano "grandezze che escono dall'ordinario".

Differenze centrali

1. **Etichettatura vs singolo ordine d'infinito.** NSA produce una gerarchia di infiniti/infinitesimi ma *non* etichetta automaticamente gli infiniti in funzione del numeratore come fa AIE2 con ∞_n . In NSA due espressioni che divergerebbero per numeratori diversi possono essere rappresentate da iperreali diversi ma non c'è la diretta e canonica corrispondenza $\frac{n}{0} \mapsto \infty_n$ costruita in AIE2.
2. **Struttura algebrica.** ${}^*\mathbb{R}$ è (nella costruzione standard) un campo esteso: conserva addizione, moltiplicazione e inversi (ad eccezione dello zero). L'insieme $\mathbb{A}$ di AIE2 (unioni finite di blocchi $\infty_n + \perp$) non è un campo né un gruppo additivo (non esiste identità additiva all'interno di $\mathcal{U}$, le inversi non esistono in genere): AIE2 è pensata come *algebra di insiemi* con regole operative insiemistiche, non come campo numerico.
3. **Fondamento costruttivo e computabilità.** Le costruzioni classiche di NSA spesso implicano assunzioni non costruttive (ultrafilter non principal) e sono meno immediate da adattare a contesti fortemente computazionali o formalizzabili in modo completamente costruttivo. AIE2 è progettata fin dall'inizio con una vocazione *computazionale* (mappa start/end, operazioni su naturali) che la rende più adatta a formalizzazioni efficaci (Lean, CAS).

Esempio comparativo

- NSA: esiste $H \in {}^*\mathbb{R}$ con H infinito (cioè per ogni $N \in \mathbb{N}$, $|H| > N$). Non esiste una étichetta canonica che lo leghi a un numeratore specifico senza ulteriori convenzioni.
- AIE2: $\frac{n}{0} = \infty_n = [n, n+1)$ mantiene esplicitamente informazione sul numeratore n .

Conclusione (NSA vs AIE2)

Scegli NSA quando si desidera: (i) una struttura di campo estesa con infinitesimi; (ii) argomentazioni analitiche basate su transfer principle. Scegli AIE2 quando si desidera: (i) conservare *etichetta numeratore* e contesto; (ii) implementare regole operative e algoritmi di regolarizzazione in modo costruttivo e verificabile.

7.3 Confronto con la sfera di Riemann e l'infinito proiettivo

Richiamo rapido

Nell'analisi complessa la sfera di Riemann $\hat{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$ aggiunge un unico punto "all'infinito" permettendo di trattare razionalmente funzioni meromorfe con un comportamento compatto.

Differenze essenziali

- **Unico punto all'infinito vs infiniti etichettati.** La sfera di Riemann ha un'unica entità ∞ . AIE2 invece sostituisce il singolo ∞ con una famiglia $\{\infty_n\}$ che sono *intervalli* distinti e portano informazione numerica diversa.
- **Topologia e compattezza vs operatività.** La sfera è uno strumento topologico/analitico (compattezza, prolungamento analitico). AIE2 è invece uno strumento simbolico-operativo: serve a mantenere e manipolare etichette e supporti numerici nelle operazioni.

Quando preferire la sfera di Riemann

- per motivi di analisi complessa, trattare poli e residui, e usare argomentazioni geometriche globali;

- quando l'identificazione di tutti i 'punti all'infinito' è concettualmente corretta e sufficiente.

Quando preferire AIE2

- quando è utile distinguere $\frac{1}{0}$ da $\frac{2}{0}$ e propagare questa distinzione in calcoli simbolici;
- nei contesti computazionali e di regolarizzazione dove l'informazione sul numeratore è rilevante.

7.4 Confronto con i metodi classici di regolarizzazione

Richiamo rapido

I metodi classici (cut-off, analitic continuation / zeta-regularization, dimensional regularization, Pauli–Villars, etc.) forniscono numeri finiti a partire da espressioni divergenti usando procedure analitiche o fisiche.

Analogie e differenze

- **Obiettivo comune:** dare senso a divergenze e assegnare valori computabili.
- **Differenza metodologica:** i metodi classici tendono a restituire *valori numerici* (es. -regularization restituisce $\zeta(-1) = -\frac{1}{12}$ per $\sum_{n \geq 1} n$ nella convenzione usuale), mentre AIE2 enfatizza la *conservazione simbolica* e l'etichettatura: la regolarizzazione in AIE2 (operatore R e funzione Φ) produce simboli regolarizzati che mantengono metadati e etichette e poi, solo se opportuno, valori numerici tramite Φ .
- **Computabilità e spiegabilità:** AIE2 richiede che R sia computabile e tracci il contesto di regolarizzazione; i metodi classici sono spesso meno esplicitamente computabili (o richiedono scelte analitiche che vanno motivate).

Esempio

- Zeta-regularization: si usa il prolungamento analitico della funzione (s) e si assegna $\sum_{n \geq 1} n \mapsto \zeta(-1) = -\frac{1}{12}$.

- AIE2: si applica R che produce un simbolo S con etichetta e ordine; $\Phi(S)$ può restituire un valore numerico che va interpretato nel contesto (la scelta di R e la motivazione della scelta sono documentate come parte della procedura).

Conclusione

I metodi classici forniscono valori numerici che risultano efficaci in molte applicazioni fisiche; AIE2 propone di integrare queste valutazioni con informazione simbolica, tracciabilità e procedure computabili che esplicitano le scelte effettuate nella regolarizzazione.

7.5 Confronto con la Logica Modale

Richiamo rapido

La logica modale introduce operatori $\Box$ (necessità) e $\Diamond$ (possibilità) per esprimere verità relative a "mondi possibili" collegati da una relazione di accessibilità (Kripke semantics).

Analogie concettuali

- Le *etichette* di AIE2 funzionano come *contesti* o mondi: un elemento x_β è valido in un contesto/etichetta β . In questo senso esiste una forte analogia concettuale (contesti vs mondi possibili).
- Alcune trasformazioni etichettali in AIE2 (morfismi $\beta \rightarrow \gamma$) giocano un ruolo simile alle relazioni di accessibilità che connettono mondi in logica modale.

Differenze formali

- La logica modale è essenzialmente una logica del linguaggio e della verità; AIE2 è una teoria algebrica/simbolica per la manipolazione di oggetti numerici estesi.
- AIE2 non introduce operatori modali per esprimere possibilità/necessità; la "modalità" informativa è codificata direttamente nelle etichette e nelle regole operative (più un approccio simbolico che logico).

Mappatura pratica

Un modo utile di pensare alla relazione è:

$$\text{etichetta } \beta \quad \longleftrightarrow \quad \text{mondo } w_\beta$$

e le trasformazioni etichettali sono funzioni tra mondi. Tuttavia, l'uso pratico differisce: in AIE2 si manipolano esplicitamente i contenuti numerici (intervalli), mentre in logica modale si ragiona su proprietà e verità.

7.6 Confronto con la Teoria di Galois (analogie strutturali)

Richiamo rapido

La teoria di Galois studia estensioni di campi e i loro gruppi di automorfismi; fornisce criteri per la risolubilità algebrica e simmetrie interne alle estensioni.

Analogia e differenze

- **Analogia:** in AIE2 esistono operatori fondamentali (inversione I , regolarizzazione R , trasformazioni etichettali) che generano una struttura formale di trasformazioni su $\mathbb{A}$. Queste trasformazioni possono essere viste come analoghe ai gruppi di automorfismi che agiscono su estensioni in Galois.
- **Differenza:** mentre la teoria di Galois è profondamente radicata in strutture di campo e polinomi, AIE2 è costruita su operazioni insiemistiche e su politiche computazionali/epistemiche; non si parla di risolubilità di polinomi nella stessa forma.

Uso proficuo dell'analogia

L'analogia è utile dal punto di vista concettuale: può guidare la definizione di *gruppi di simmetria* interni ad AIE2 (automi che preservano etichette o che scambiano classi di etichette compatibili). Questo può portare a studi di invarianza di regole di regolarizzazione rispetto a trasformazioni di etichette.

7.7 Vantaggi e limiti comparativi di AIE2

Vantaggi di AIE2

- **Tracciabilità simbolica:** AIE2 conserva etichette e metadati che consentono di ricostruire la storia simbolica di una regolarizzazione o di una semplificazione.
- **Implementabilità:** la mappa start/end e le formule sugli estremi rendono l'implementazione efficace e facilmente formalizzabile in proof assistants (Lean) e in CAS.

- **Chiarezza operativa:** le regole sono insiemistiche e concrete — utili per chi deve ragionare su impatti pratici (ingegneria, informatica, fisica computazionale).
- **Flessibilità applicativa:** la scelta di politiche di propagazione e di compressione rende la teoria adattabile a casi d'uso differenti.

Limiti e punti di attenzione

- **Mancanza di struttura di campo:** $\mathbb{A}$ non è un campo; molte tecniche analitiche che richiedono questa struttura non sono immediatamente disponibili.
- **Scelte arbitrarie:** alcune decisioni (propagazione forte di $\perp$, rappresentante unico) sono convenzionali e devono essere motivate caso per caso; ciò può portare a differenti istanze pratiche della teoria.
- **Interoperabilità con analisi classica:** passare da $\mathbb{A}$ a risultati reali classici richiede la definizione chiara di Φ e di quando andare da un simbolo regolarizzato a un valore numerico.

7.8 Raccomandazioni pratiche: quando usare AIE2

- **Progetti computazionali e formalizzazione:** preferire AIE2 quando si deve implementare o formalizzare regole di manipolazione simbolica e regolarizzazione in modo verificabile (es. Lean, motori CAS).
- **Analisi qualitativa e diagnostica:** usare AIE2 per tracciare "da dove viene l'infinito" (numeratore), per fare diagnosi in modelli fisici o numerici e per definire politiche di fallback.
- **Analisi rigorosa e teoretica:** se l'obiettivo è costruire una teoria di campo con infinitesimi e sfruttare transfer principle, NSA o approcci analitici classici (sfera di Riemann, prolungamento analitico) potrebbero essere più appropriati.

7.9 Indicazioni editoriali per il libro

Per rendere questa sezione efficace nel libro, si suggerisce di:

1. includere brevi *box* comparativi (pro/contro) per ogni teoria concorrente;

2. fornire esempi paralleli risolti sia in AIE2 che nell'altro formalismo (es. come affrontare $1/0$, o la serie divergente $\sum n$), così da mostrare i trade-off pratici;
3. aggiungere riferimenti standard per approfondire: testi su NSA (Robinson), su sfera di Riemann e teoria delle funzioni, su zeta-regularization e regolarizzazione fisica, su Galois theory e logica modale.

7.10 Riepilogo del capitolo

Riassunto

Cosa ricordare:

- AIE2 è uno strumento operativo e computazionale che differisce nettamente da costruzioni algebriche (NSA) o topologiche (sfera di Riemann) pur condividendone l'intento di trattare infiniti/infinitesimi.
- I punti di forza di AIE2 sono la tracciabilità delle etichette, la facilità di formalizzazione e l'applicabilità pratica; i limiti principali sono l'assenza di una struttura di campo e la necessità di giustificare scelte convenzionali nelle politiche di regolarizzazione.
- Nel libro si raccomanda di affiancare a questa analisi esempi paralleli e riferimenti bibliografici per guidare il lettore nella scelta dello strumento più adatto al proprio problema.

7.11 Esempi paralleli e riferimenti bibliografici

Esempi paralleli

Esempio 1: Divisione per zero ($1/0$).

- **AIE2:** $\frac{1}{0} = \infty_1 = [1, 2)$. L'etichetta conserva informazione sul numeratore e produce un oggetto intervallare manipolabile nelle regole operative.
- **NSA:** in ${}^*\mathbb{R}$ non esiste un elemento "canonico" corrispondente a $1/0$. La divisione è indefinita, salvo estensioni sintattiche non standard. L'idea è che nessun iperreale finito moltiplicato per 0 dia 1.
- **Sfera di Riemann:** $1/0 = \infty$, unico punto all'infinito. Non si distingue da $2/0$.

Esempio 2: Serie divergente $\sum_{n=1}^{\infty} n$.

- **AIE2:** la serie non converge nei reali ma può essere interpretata come unione di intervalli etichettati generati da $\frac{n}{0}$. Con un operatore di regolarizzazione R e una funzione Φ , si può associare a questo simbolo un valore finito tracciabile.
- **Zeta-regularization:** tramite prolungamento analitico $\zeta(-1) = -\frac{1}{12}$.
- **Metodi fisici (cut-off):** introducendo un limite N , si calcola $\sum_{n=1}^N n = \frac{N(N+1)}{2}$, poi si sottrae un termine di divergenza e si procede al limite $N \rightarrow \infty$.

Esempio 3: Indeterminato ($0/0$).

- **AIE2:** $\frac{0}{0} = \perp$, un oggetto distinto che propaga fortemente nelle operazioni.
- **Analisi classica:** indeterminato, può assumere valori diversi a seconda del limite considerato.
- **NSA:** si possono considerare rapporti di infinitesimi $\frac{\varepsilon}{\varepsilon}$ che valgono 1, ma non c'è un singolo simbolo corrispondente a $0/0$.

Riferimenti bibliografici essenziali

Per il lettore che desidera approfondire i confronti, suggeriamo i seguenti testi di riferimento:

- **Analisi Non Standard (NSA):**
 - A. Robinson, *Non-standard Analysis*, North-Holland, 1966.
 - K. Hrbacek, O. Lessmann, R. O. Duncan, *Analysis with Ultrasmall Numbers*, Springer, 2014.
- **Sfera di Riemann e analisi complessa:**
 - L. Ahlfors, *Complex Analysis*, McGraw-Hill, 1979.
 - J. Conway, *Functions of One Complex Variable*, Springer, 1978.
- **Metodi di regolarizzazione in fisica:**
 - C. Itzykson, J.-B. Zuber, *Quantum Field Theory*, McGraw-Hill, 1980.
 - K. Hepp, *Quantum Theory of Many-Particle Systems*, Springer, 2005.
- **Logica Modale:**

- B. Chellas, *Modal Logic: An Introduction*, Cambridge Univ. Press, 1980.
- P. Blackburn, M. de Rijke, Y. Venema, *Modal Logic*, Cambridge Univ. Press, 2001.

- **Teoria di Galois:**

- I. Stewart, *Galois Theory*, Chapman and Hall, 2004.
- H. M. Edwards, *Galois Theory*, Springer, 1984.

Riassunto

Gli esempi paralleli aiutano a capire in pratica cosa cambia passando da una teoria all'altra, mentre i riferimenti bibliografici permettono al lettore di approfondire in autonomia. L'obiettivo non è sostituire le altre teorie ma mostrare come AIE2 possa convivere e interagire con esse, fornendo una prospettiva nuova.

Chapter 8

Interpretazioni filosofiche ed epistemiche

Riassunto

In sintesi: in questo capitolo analizziamo AIE2 dal punto di vista filosofico ed epistemico. Il problema non è solo tecnico (gestire infiniti), ma anche concettuale: come giustificare nuove entità matematiche? Che significato hanno in termini di conoscenza, costo epistemico e politiche di interpretazione?

8.1 Premessa epistemologica

Ogni nuova teoria matematica non introduce soltanto regole operative, ma modifica il *quadro epistemico* con cui pensiamo concetti fondamentali come infinito, indeterminato e regolarizzazione.

In particolare:

- L'Analisi Classica si fonda sull'idea che ogni espressione abbia un valore ben definito nei reali o, se non lo ha, si dichiari "indefinita".
- L'Analisi Non Standard (NSA) introduce nuovi oggetti (iperreali) giustificandoli con principi di estensione logica (ultraprodotti, transfer principle).
- AIE2 propone invece un approccio *operativo-etichettale*: non mira a un campo esteso, ma a un insieme computabile di simboli con regole tracciabili.

Intuizione

L'originalità epistemica di AIE2 sta nell'assumere che la divisione per zero non debba essere annullata come "errore", ma reinterpretata come un'occasione per produrre informazione (un'etichetta).

8.2 Il concetto di costo epistemico

Definizione

Definizione 8.1. Chiamiamo *costo epistemico* l'onere aggiuntivo di conoscenza richiesto da un formalismo rispetto a un altro. Un sistema matematico ha basso costo epistemico se le sue regole sono facilmente apprendibili, spiegabili e riconducibili a concetti intuitivi.

Costo epistemico di AIE2

- **Basso costo operativo:** le regole insiemistiche (unioni di intervalli, somme di Minkowski) sono concetti familiari a studenti e ricercatori.
- **Costo di astrazione:** il lettore deve abituarsi a pensare $\frac{n}{0}$ non come infinito, ma come intervallo etichettato. Questo è un cambio concettuale ma non troppo oneroso.
- **Alto beneficio cognitivo:** si guadagna in tracciabilità e capacità diagnostica: ogni infinito porta con sé un'etichetta che racconta la sua "origine".

Esempio 8.1. In Analisi Classica, $\frac{1}{0}$ è "infinito". In AIE2, $\frac{1}{0} = \infty_1 = [1, 2)$. Il costo epistemico consiste nell'accettare una nuova convenzione, ma in cambio si ottiene un oggetto calcolabile e tracciabile.

8.3 Politiche di propagazione e giustificazione

AIE2 introduce alcune scelte convenzionali che richiedono giustificazione:

1. Propagazione dell'indeterminato

$$x + \perp = \perp, \quad x \cdot \perp = \perp.$$

Intuizione

Questa regola riflette un principio epistemico: se una parte del calcolo è priva di significato (indeterminata), l'intero risultato non deve essere considerato affidabile.

2. Compressione delle unioni di intervalli

$$[m, m+1) \cup [m+1, m+2) = [m, m+2).$$

Intuizione

Questa scelta riduce complessità e facilita la rappresentazione. Ha però un costo epistemico: si perde l'informazione che il risultato proveniva da due etichette distinte.

Attenzione

Qui emerge un punto critico: maggiore compressione = meno tracciabilità. Il progettista o il matematico deve decidere quanto privilegiare la semplicità rispetto alla conservazione dell'informazione.

8.4 Interpretazioni filosofiche dell'infinito etichettato**1. Interpretazione costruttivista**

Gli infiniti etichettati sono visti come *strumenti operativi* che nascono da regole definite e non da assiomi metafisici. Un costruttivista accetterebbe AIE2 perché tutto è computabile e basato su definizioni finite.

2. Interpretazione platonista

Un platonista potrebbe leggere gli ∞_n come entità realmente esistenti, “blocchi” dell'infinito preesistenti alla mente umana. In questa prospettiva, AIE2 non inventa ma scopre una nuova struttura dell'infinito.

3. Interpretazione pragmatico-strumentale

Dal punto di vista applicativo, non importa se ∞_n “esiste” davvero. Conta che fornisca un framework utile per regolarizzare calcoli divergenti, con tracciabilità e spiegabilità.

8.5 Ricadute metodologiche

- **Per la matematica pura:** AIE2 propone una nuova algebrica dell'infinito che può aprire strade a nuove categorie di strutture (insiemi intervallari con etichette).
- **Per la fisica teorica:** AIE2 fornisce una grammatica simbolica per regolarizzare divergenze mantenendo traccia dell'origine dei termini infiniti.
- **Per l'informatica:** la natura computabile delle regole rende AIE2 adatta a implementazioni in Lean, Coq, CAS o simulazioni numeriche.
- **Per la filosofia della scienza:** AIE2 diventa un caso di studio su come le convenzioni e le regole operative modellino la conoscenza matematica.

8.6 Riepilogo del capitolo

Riassunto

Cosa ricordare:

1. Il concetto chiave è il *costo epistemico*: accettare nuove regole comporta un piccolo prezzo cognitivo, ma porta grandi benefici in tracciabilità e calcolabilità.
2. Le scelte operative (propagazione di $\perp$, compressione di intervalli) non sono solo tecniche ma anche epistemiche: implicano compromessi tra semplicità e conservazione di informazione.
3. Le interpretazioni dell'infinito etichettato possono essere costruttiviste, platoniste o strumentali: AIE2 resta valido in tutti i quadri.
4. Le ricadute metodologiche vanno oltre la matematica, toccando fisica, informatica e filosofia della scienza.

Chapter 9

Applicazioni di AIE2

Riassunto

In sintesi: presentiamo applicazioni concrete di AIE2 in matematica, fisica teorica, informatica e economia/finanza. Ogni sezione segue lo stesso schema operativo: *modellazione* $\rightarrow$ *traduzione in AIE2* $\rightarrow$ *calcolo con le regole* $\rightarrow$ *regolarizzazione* $(R, \Phi) \rightarrow$ *interpretazione e verifica*. Gli esempi sono completi, senza diagrammi, e terminano con una *Forma Normale* (FN) e un commento interpretativo.

9.1 Guida rapida al workflow applicativo

Dato un problema che produce divergenze o divisioni per zero, seguiamo i passi:

1. **Modellazione (dominio di partenza).** Scrivere l'espressione originaria (analitica, numerica, algoritmica o econometrica) e individuare i punti di divergenza.
2. **Traduzione in AIE2.** Sostituire le occorrenze di $\frac{n}{0}$ con $\infty_n = [n, n + 1)$; mappare ogni $\frac{0}{0}$ in $\perp$. Per espressioni più complesse, introdurre variabili $U, V, \dots \in \mathcal{U} \subset \mathbb{A}$ secondo i criteri del nucleo.
3. **Calcolo simbolico-operativo.** Applicare le regole sugli estremi: $[a, b) \oplus [c, d) = [a + c, b + d)$, $k \cdot [a, b) = [ka, kb)$, $[a, b) \odot [c, d) = [ac, bd)$, con compressione canonica delle unioni in intervalli.
4. **Regolarizzazione (R, Φ) .** Scegliere un operatore R (documentato) che invii $[p, q) \mapsto S_{p,q}$ in una famiglia di simboli regolarizzati, e un funzionale Φ che, se desiderato, produca un numero finito $\Phi(S_{p,q})$.
5. **Interpretazione e verifica.** Discutere la scelta di R (costo epistemo, trasparenza), confrontare con metodi classici/alternativi, validare

sul problema originario.

Intuizione

La forza di AIE2 è la *tracciabilità*: l'informazione sul numeratore n non va persa, ma viene propagata nelle operazioni, consentendo scelte di regolarizzazione motivate e riproducibili.

9.2 Matematica: limiti, poli e serie divergenti

9.2.1 Poli semplici e divisione per zero

Consideriamo $f(x) = \frac{n}{x}$ e $x \rightarrow 0$. In analisi classica il limite non esiste (diverge). In AIE2, nel momento in cui si idealizza l'operatore “valuta in zero” come sostituzione simbolica, si ottiene $\frac{n}{0} = \infty_n = [n, n+1)$.

Esempio 9.1 (Somma di poli etichettati). Sia $g(x) = \frac{1}{x} + \frac{2}{x}$. In AIE2, alla sostituzione formale:

$$\frac{1}{0} \oplus \frac{2}{0} = \infty_1 \oplus \infty_2 = [1, 2) \oplus [2, 3) = [3, 5).$$

FN: $[3, 5) = \infty_3 \cup \infty_4$. **Lettura:** la divergenza aggregata conserva l'informazione dei numeratori.

9.2.2 Prodotti di divergenze

Esempio 9.2 (Prodotto di due poli).

$$\left(\frac{1}{0}\right) \odot \left(\frac{3}{0}\right) = \infty_1 \odot \infty_3 = [1, 2) \odot [3, 4) = [3, 8).$$

FN: $[3, 8) = \infty_3 \cup \dots \cup \infty_7$. **Interpretazione:** il prodotto amplia il supporto d'infinito.

9.2.3 Serie divergenti e zeta-regolarizzazione

Esempio 9.3 (Serie di ordine uno). La serie $\sum_{k=1}^{\infty} k$ può essere trattata in due modi complementari:

- **Zeta-regolarization classica:** per prolungamento analitico, $\sum k \mapsto \zeta(-1) = -\frac{1}{12}$.
- **AIE2:** rappresentiamo il contributo divergente come simbolo interval-lare U costruito a partire dai termini che “spingono all'infinito”. Appli-chiamo $R : [p, q) \mapsto S_{p,q}$ e poi $\Phi(S_{p,q}) \in \mathbb{R}$ se serve un numero.

Scelta esemplificativa: $R([p, q]) :=$ “etichetta condensata” $S_{p,q}$ e $\Phi(S_{p,q}) := \frac{p+q-1}{2}$ (centroide discreto). Allora la serie sopra, modellata come $U = [p, q]$ con p grande, produce un numero rappresentativo $\Phi(U)$ che documenta *quale* intervallo di infiniti è stato coinvolto. Questa scelta è trasparente e riproducibile.

9.2.4 Confronto operativo

- Metodi classici restituiscono direttamente un *numero*.
- AIE2 restituisce prima un *simbolo etichettato* (FN), poi eventualmente un numero via (R, Φ) .

9.3 Fisica teorica: contributi divergenti e tracciabilità

9.3.1 Somma di contributi a un’ampiezza

Sia $\mathcal{A} = \mathcal{A}_1 + \mathcal{A}_2$ dove ciascun contributo diverge come $\frac{n_i}{0}$. In AIE2, traduciamo $\mathcal{A}_i \mapsto \infty_{n_i}$.

Esempio 9.4 (Ampiezza a due contributi).

$$\mathcal{A} \rightsquigarrow \infty_{n_1} \oplus \infty_{n_2} = [n_1, n_1 + 1) \oplus [n_2, n_2 + 1) = [n_1 + n_2, n_1 + n_2 + 2).$$

FN: $\bigcup_{k=n_1+n_2}^{n_1+n_2+1} \infty_k$. **Interpretazione:** la localizzazione dell’infinito aggregato dipende direttamente dalle etichette n_1, n_2 .

9.3.2 Propagazione dell’indeterminato

Esempio 9.5 (Canale indeterminato). Se un canale contiene una struttura $\frac{0}{0}$, traduciamo $\perp$ e

$$\mathcal{A}_{\text{tot}} = (\infty_n \oplus \perp) \odot \infty_m = \perp.$$

Lettura: principio conservativo; il risultato è epistemicamente inaffidabile senza regolarizzazione dedicata.

9.3.3 Regolarizzazione documentata

Definizione 9.1 (Scelta di R in fisica). Una politica naturale è: $R([p, q]) :=$ simbolo $S_{p,q}^{(\lambda)}$ dove λ codifica lo schema (cut-off, dimensionale, Pauli–Villars). Definire $\Phi_\lambda(S_{p,q}^{(\lambda)})$ in modo coerente con lo schema adottato, mantenendo traccia di (p, q, λ) .

Intuizione

AIE2 non “sostituisce” i metodi fisici: li incapsula in un livello simbolico che conserva *da dove* proviene la divergenza.

9.4 Informatica: overflow, static analysis e calcolo affidabile

9.4.1 Overflow e NaN semantici

Nei calcoli floating point, Inf e NaN perdono spesso il contesto. Con AIE2:

$$\text{Inf}(n) \equiv \infty_n, \quad \text{NaN} \equiv \perp.$$

Esempio 9.6 (Somma con overflow etichettato). In un motore numerico, `overflow(1)` e `overflow(2)` propagano come

$$\infty_1 \oplus \infty_2 = [3, 5),$$

consentendo all’orchestratore di decidere *policy* diverse se il supporto supera una soglia (*e.g.* attiva fallback).

9.4.2 Dominio astratto per analisi statica

Definizione 9.2 (Dominio astratto AIE2). Definiamo il dominio $\mathcal{D}_{\text{AIE2}} = \{[p, q)\} \cup \{\perp\}$ parzialmente ordinato per inclusione degli intervalli e con $\perp$ come elemento *top* (recapito dell’incertezza massima). La join $\sqcup$ è l’unione con compressione, il meet $\sqcap$ può essere l’intersezione (se vuota, si ritorna a una politica scelta: $\perp$ o $\perp$).

Esempio 9.7 (Astrazione di un blocco di codice aritmetico). Supponiamo che un ramo possa produrre $\frac{1}{0}$ e l’altro $\frac{3}{0}$:

$$\text{join} : \infty_1 \sqcup \infty_3 = [1, 2) \cup [3, 4) = [1, 2) \cup [3, 4) \quad (\text{non contigui}).$$

Se si attiva una compressione *aggressiva* a singolo intervallo, si può mappare $[1, 2) \cup [3, 4) \mapsto [1, 4)$. **Trade-off:** semplicità vs granularità d’informazione.

9.4.3 Regole operative in un runtime

Regola R1 (somma): $[a, b) \oplus [c, d) = [a + c, b + d)$. **R2 (prodotto):** $[a, b) \odot [c, d) = [ac, bd)$. **R3 (indeterminato):** qualsiasi operazione con $\perp$ produce $\perp$.

Queste regole, implementate in un runtime, consentono di eseguire calcoli robusti senza perdere il contesto delle anomalie.

9.5 Economia e finanza: rapporti, elasticità, volatilità

9.5.1 Crescita con baseline nulla

Il tasso di crescita percentuale $g = \frac{P_1 - P_0}{P_0}$ è problematico se $P_0 = 0$.

Esempio 9.8 (Crescita da zero). Se $P_0 = 0$ e $P_1 = n$, allora $g = \frac{n}{0} = \infty_n = [n, n+1)$. **FN:** $[n, n+1)$. **Interpretazione:** la crescita è “infinitamente grande” ma *etichettata* dal livello n . Nell’analisi comparata tra prodotti, confrontare ∞_n e ∞_m equivale a confrontare gli intervalli $[n, n+1)$ e $[m, m+1)$.

9.5.2 Elasticità di prezzo

L’elasticità $E = \frac{\Delta Q/Q}{\Delta P/P}$ diventa problematica se $\Delta P = 0$ o $P = 0$.

Esempio 9.9 (Elasticità “infinita”). Per $\Delta P = 0$ e $\Delta Q \neq 0$:

$$E = \frac{\Delta Q/Q}{0} = \infty_n \quad \text{con } n = \left\lceil \left| \frac{\Delta Q}{Q} \right| \cdot \kappa \right\rceil,$$

dove $\kappa > 0$ è un fattore di scalatura scelto come politica. **Lettura:** si ottiene un’etichetta confrontabile tra mercati.

9.5.3 Sharpe ratio e varianza quasi nulla

Lo Sharpe $S = \frac{\mathbb{E}[R] - r_f}{\sigma(R)}$ diverge se $\sigma(R) \rightarrow 0^+$ con numeratore positivo.

Esempio 9.10 (Sharpe etichettato). Se empiricamente $\sigma \approx 0$ e $\mu := \mathbb{E}[R] - r_f = n > 0$, allora $S \rightsquigarrow \frac{n}{0} = \infty_n$. Nell’aggregazione di portafogli:

$$\infty_{n_1} \oplus \infty_{n_2} = [n_1 + n_2, n_1 + n_2 + 2),$$

consentendo di documentare l’origine delle eccezioni e applicare una regola R (es. capping, winsorization etichettata).

9.6 Scelte di (R, Φ) per dominio

Attenzione

Trasparenza: ogni scelta di R e Φ va documentata. In particolare, riportare: motivazione, parametri, impatti sulle decisioni e sensibilità a variazioni (*stress test* epistemico).

Dominio	Esempio di R	Esempio di Φ
Matematica	$R([p, q]) = S_{p,q}^{(\text{cent})}$	$\Phi(S) = \frac{p+q-1}{2}$ (centroide discreto)
Fisica	$R([p, q]) = S_{p,q}^{(\lambda)}$	Φ_λ coerente con lo schema (cut-off, dimensionale, Pauli–Villars)
Informatica	$R([p, q]) = S_{p,q}^{(\text{cap}, M)}$	$\Phi(S) = \min\{q-1, M\}$ (capping controllato)
Economia/Finanza	$R([p, q]) = S_{p,q}^{(\text{rank})}$	$\Phi(S) = \text{rank}(p, q)$ per ordinare casi estremi

Table 9.1: Esempi di scelte (R, Φ) per diversi domini applicativi.

9.7 Casi studio completi

9.7.1 Caso studio A: razionale con doppio polo all'origine

Sia

$$F(x) = \frac{1}{x} + \alpha \frac{2}{x} + \beta \frac{3}{x^2} \quad (\alpha, \beta \in \mathbb{R}).$$

Traduzione AIE2 (valutazione formale in $x = 0$):

$$F(0) \rightsquigarrow \infty_1 \oplus \alpha \cdot \infty_2 \oplus \beta \cdot (\infty_3 \odot \infty_1).$$

Calcolo: $\alpha \cdot \infty_2 = [2\alpha, 2\alpha + \alpha]$ (nota: usare $k \cdot [a, b] = [ka, kb]$; per $k \in \mathbb{R}^+$ si assume compatibilità con la versione estesa continua); $\infty_3 \odot \infty_1 = [3, 4] \odot [1, 2] = [3, 8]$.

$$\Rightarrow F(0) \rightsquigarrow [1, 2] \oplus [2\alpha, 3\alpha] \oplus \beta \cdot [3, 8] = [1 + 2\alpha + 3\beta, 2 + 3\alpha + 8\beta].$$

FN: intervallo unico; **Interpretazione:** i parametri α, β traslano ed espandono il supporto dell'infinito.

9.7.2 Caso studio B: pipeline numerica con fallback

Una pipeline calcola $y = (a/b) + (c/d)$. Se $b = 0$ o $d = 0$ il sistema attiva fallback.

$$y \rightsquigarrow \begin{cases} \infty_a \oplus \infty_c, & b = d = 0, \\ \infty_a \oplus \frac{c}{d}, & b = 0, d \neq 0, \\ \frac{a}{b} \oplus \infty_c, & b \neq 0, d = 0. \end{cases}$$

Politica: se la FN contiene ∞_k con $k > K$ (soglia), si esegue R con capping: $R([p, q]) = S_{p, \min(q-1, K)}$. **Effetto:** l'output mantiene l'etichetta e consente auditing a posteriori.

9.7.3 Caso studio C: indicatori economici estremi

Indice $I = \frac{\Delta \text{vendite}}{\text{costo unitario}}$ con costo unitario talvolta nullo per promozioni.

$$I \rightsquigarrow \frac{n}{0} = \infty_n, \quad n = \Delta \text{vendite}.$$

Rendicontazione: ordinare i casi per n (o $[n, n+1)$) e applicare Φ di ranking per comparazioni inter-negozio.

9.8 Checklist di validazione

1. **Mappatura corretta:** ogni divisione per zero è stata tradotta in ∞_n o $\perp$ in modo coerente?
2. **Regole applicate:** somma/prodotto/scalatura rispettano le formule sugli estremi e la compressione?
3. **FN esplicita:** ogni risultato è riportato in Forma Normale e, se utile, come unione di generatori?
4. **Scelta** (R, Φ) : documentata, motivata, con parametri e analisi di sensibilità?
5. **Confronto:** è stato mostrato un confronto con un metodo alternativo (classico/industriale) per triangolare i risultati?
6. **Auditabilità:** le etichette ∞_n sono tracciate fino ai dati/sorgenti originali?

9.9 Esercizi con soluzioni

Esercizi

1. **(Matematica)** Calcolare in FN: $(\infty_1 \oplus 2 \cdot \infty_2) \odot \infty_3$.
2. **(Fisica)** Un'ampiezza ha tre contributi divergenti $\frac{1}{0}, \frac{2}{0}, \frac{4}{0}$. Scrivere la FN della somma e proporre un R coerente con un cut-off a soglia K .
3. **(Informatica)** Una funzione può restituire ∞_2 o $\perp$. Calcolare la *join* nel dominio $\mathcal{D}_{AIE2}$ e discutere due politiche: (i) $\perp$ top assorbente, (ii) top differito.
4. **(Economia)** Uno Sharpe empirico produce ∞_5 . Un secondo indice, indipendente, produce ∞_3 . Qual è la FN dell'aggregazione? Proporre Φ che restituisca un ranking stabile.

Soluzioni

1. $\infty_1 \oplus 2 \cdot \infty_2 = [1, 2) \oplus [4, 6) = [5, 8)$. Prodotto con $\infty_3 = [3, 4)$: $[5, 8) \odot [3, 4) = [15, 32)$. **FN:** $\infty_{15} \cup \dots \cup \infty_{31}$.

2. Somma: $[1, 2) \oplus [2, 3) \oplus [4, 5) = [7, 10)$. Con cut-off K , porre $R([p, q)) = S_{p, \min(q-1, K)}$. $\Rightarrow R([7, 10)) = S_{7, \min(9, K)}$.
3. (i) Con $\perp$ top assorbente: $\infty_2 \sqcup \perp = \perp$. (ii) Top differito: mantenere coppia $(\infty_2, \perp)$ fino a disambiguazione; se non possibile, collassare a $\perp$.
4. Aggregazione: $[5, 6) \oplus [3, 4) = [8, 10)$. Una possibile Φ di ranking è $\Phi([p, q)) = (p, q)$ ordinato lessicograficamente (prima p , poi q).

9.10 Riepilogo del capitolo

Riassunto

Cosa ricordare:

- AIE2 fornisce una via *operativa e tracciabile* per trattare divergenze in domini differenti.
- La conservazione delle etichette ∞_n consente auditing, confronto tra metodi e scelte di regolarizzazione informate.
- Le politiche (R, Φ) sono il ponte verso numeri finiti e decisioni pratiche: vanno scelte e documentate per dominio.

Chapter 10

Implementazione computazionale di AIE2

Riassunto

In sintesi: questo capitolo fornisce una specifica tecnica completa per implementare AIE2 in un CAS, in un linguaggio general-purpose (es. Python/Julia) e in un proof assistant (es. Lean). Definiamo i tipi di dato, gli algoritmi di forma normale (FN), la semantica operativa di somma, prodotto e scalatura, le politiche di compressione, la propagazione di $\perp$, la struttura plug-in per (R, Φ) , le procedure decisionali e i test di conformità.

10.1 Obiettivi e scelte di design

- **Tracciabilità:** ogni oggetto “infinito” è un intervallo $[n, n+1)$ (con $n \in \mathbb{N}_{\geq 1}$) o unione di tali intervalli; l’origine dei contributi deve rimanere accessibile fino all’esito finale (log interno).
- **FN canonica:** ogni risultato è riportato in *Forma Normale* come unione *disgiunta* e (opzionalmente) *compressa* in intervalli contigui, secondo una *politica* esplicita.
- **Determinismo:** le operazioni producono risultati univoci a parità di politica scelta.
- **Modularità:** (R, Φ) è un plug-in; l’algebra di base non dipende da particolari scelte di regolarizzazione.

10.2 Modello dei dati

10.2.1 Tipi principali

Definizione 10.1 (Generatori). Un *generatore* è l'intervallo semiaperto $\infty_n := [n, n + 1)$ con $n \in \mathbb{N}_{\geq 1}$.

Definizione 10.2 (Oggetti AIE2). Sia $\mathcal{G} = \{\infty_n \mid n \in \mathbb{N}_{\geq 1}\}$. Definiamo:

$$\mathbb{A} = \left\{ U \mid U = \bigcup_{i=1}^k I_i, I_i = [a_i, b_i), a_i, b_i \in \mathbb{N}, a_i < b_i \right\} \cup \{\perp\}.$$

Per default, assumiamo $a_i \geq 1$. (*Estensioni ai negativi in §??.*)

Definizione 10.3 (Forma Normale (FN)). Un $U \in \mathbb{A}$ è in *FN* se:

1. Ogni componente è un intervallo $[a, b)$ con $a, b \in \mathbb{N}$, $a < b$.
2. Gli intervalli sono disgiunti: $[a_i, b_i) \cap [a_j, b_j) = \emptyset$ per $i \neq j$.
3. (**Opzionale**) *Compressione contigua*: se $[u, v) \cup [v, w) \subseteq U$, allora in FN compaia $[u, w)$.

10.2.2 Rappresentazione concreta

- **CAS/GPL**: lista ordinata di coppie intere $[(a_1, b_1), \dots, (a_k, b_k)]$ oppure **Top** per $\perp$.
- **Lean**: tipo induttivo con costruttori **Top** | **U** (**1** : **List** (**Nat**×**Nat**)), con invarianti (ordinamento, disgiunzione).

10.3 Algoritmi di Forma Normale

10.3.1 Normalizzazione di una unione arbitraria

[FN-merge (unioni)]

Input: multinsieme di intervalli interi semiaperti $\{[a_i, b_i)\}_{i=1}^m$.

Output: lista ordinata disgiunta (e se attivo, compressa) $[(A_1, B_1), \dots, (A_k, B_k)]$.

1. Scarta ogni intervallo vuoto ($a_i \geq b_i$).
2. Ordina per a_i crescente (tie-break per b_i crescente).
3. Inizializza **out** = $\emptyset$.
4. Per ciascun $[a, b)$ in ordine:
 - Se **out** = $\emptyset$, aggiungi $[a, b)$.

- Altrimenti, sii $[A, B)$ l'ultimo. Se `compress = true` e $a \leq B$ (contigui o sovrapposti), sostituisci l'ultimo con $[A, \max(B, b))$. Se `compress = false` e $a < B$ (sovrapposizione), *merge obbligato* a $[A, \max(B, b))$ per garantire disgiunzione. Se $a = B$ e `compress = false`, aggiungi $[a, b)$ come blocco separato (contiguità non compressa).

Complessità. Ordinamento $O(m \log m)$, merge lineare $O(m)$. Totale $O(m \log m)$.

10.3.2 Conversione da generatori

Dato un set di generatori $\{\infty_{n_j}\}$, applicare FN-merge a $\{[n_j, n_j + 1)\}$.

10.4 Semantica operativa delle operazioni

Nel seguito, per chiarezza, consideriamo intervalli con estremi interi non negativi e *coefficiente scalare* $k \in \mathbb{N}$. Estensioni in §??.

10.4.1 Somma ($\oplus$)

Definizione 10.4 (Somma di Minkowski).

$$[a, b) \oplus [c, d) := [a + c, b + d).$$

Per insiemi $U = \bigcup_i [a_i, b_i)$, $V = \bigcup_j [c_j, d_j)$: $U \oplus V := \bigcup_{i,j} ([a_i, b_i) \oplus [c_j, d_j))$, poi FN.

10.4.2 Prodotto ($\odot$)

Definizione 10.5 (Prodotto positivo). Per estremi non negativi:

$$[a, b) \odot [c, d) := [a \cdot c, b \cdot d).$$

Estendere per unioni via distributività su unioni finite e FN finale.

Attenzione

Per intervalli con segni misti è necessaria la formula completa del prodotto di intervalli ($\min\{ac, ad, bc, bd\}, \max\{\cdot\}$); cfr. §??.

10.4.3 Scalatura ($k \cdot U$)

Definizione 10.6 (Scalatura per $k \in \mathbb{N}$).

$$k \cdot [a, b] := [ka, kb], \quad k \cdot \left(\bigcup_i [a_i, b_i] \right) := \bigcup_i [ka_i, kb_i].$$

FN al termine.

10.4.4 Propagazione di $\perp$

Definizione 10.7 (Top assorbente). Per ogni $U \in \mathbb{A}$:

$$U \oplus \perp = \perp, \quad U \odot \perp = \perp, \quad k \cdot \perp = \perp.$$

10.4.5 Unione e intersezione

Definizione 10.8. $\text{Union}(U, V) := \text{FN-merge}(U \cup V)$. $\text{Inter}(U, V) :=$ intersezione calcolata intervallo-per-intervallo, poi FN.

10.5 Politiche di compressione e granularità

- **compress=false** (granulare): mantiene blocchi contigui separati; massima tracciabilità, più pesante in memoria.
- **compress=true** (compatta): unisce contiguità $[u, v] \cup [v, w] \mapsto [u, w]$; minor tracciabilità, FN più semplice.
- **compress=threshold(T)**: compatta solo se $w - u \leq T$; ibrido.

Intuizione

La scelta influenza il *costo epistemico*: più compressione $\Rightarrow$ meno informazione fine; meno compressione $\Rightarrow$ più verbosità.

10.6 Struttura plug-in per (R, Φ)

10.6.1 Interfacce

Definizione 10.9 (Interfaccia R). Un R è una funzione totale $R : \mathbb{A} \rightarrow \mathbb{S}$ (spazio dei simboli regolarizzati) che rispetta:

1. **Monotonia (opzionale)**: $U \subseteq V \Rightarrow R(U) \preceq R(V)$ per un ordine in $\mathbb{S}$.

2. Compatibilità con FN: $R(U) = R(\text{FN}(U))$.

Definizione 10.10 (Interfaccia Φ). Un funzionale $\Phi : \mathbb{S} \rightarrow X$ (tipicamente $X = \mathbb{R}$ o $\mathbb{Q}$), documentato (parametri, assunzioni).

10.6.2 Esempi

- **Centroide discreto:** $R([p, q]) = S_{p,q}^{(c)}$, $\Phi(S_{p,q}^{(c)}) = \frac{p+q-1}{2}$.
- **Capping con soglia:** $R([p, q]) = S_{p, \min(q-1, K)}^{(K)}$, Φ identità sugli estremi.
- **Schema fisico λ :** $R([p, q]) = S_{p,q}^{(\lambda)}$, Φ_λ definita dal metodo (cut-off, dimensionale, ecc.).

10.7 Procedure decisionali

10.7.1 Uguaglianza

Due oggetti U, V sono uguali se, in FN con la stessa politica, le liste di intervalli coincidono esattamente.

10.7.2 Inclusione e ordine

$[a, b] \subseteq [c, d]$ sse $c \leq a$ e $b \leq d$. Estendere per unioni per verifica intervallo-per-intervallo.

10.7.3 Vuotezza

Un oggetto è vuoto se la lista di intervalli è vuota (non esiste ∞_0); $\perp$ non è vuoto ed è *top*.

10.8 Proprietà algebriche (con ipotesi)

Assumendo intervalli con estremi ≥ 0 e politiche di FN fissate:

Proposizione 10.1 (Associatività della somma). $(U \oplus V) \oplus W = U \oplus (V \oplus W)$.

Proposizione 10.2 (Commutatività della somma). $U \oplus V = V \oplus U$.

Proposizione 10.3 (Distributività su unione finita). $U \oplus (V \cup W) = (U \oplus V) \cup (U \oplus W)$ prima di FN; la FN finale preserva l'uguaglianza insiemistica.

Proposizione 10.4 (Monotonia). *Se $U \subseteq V$ allora $U \oplus W \subseteq V \oplus W$ e, per $k \geq 0$, $k \cdot U \subseteq k \cdot V$.*

Attenzione

La distributività del *prodotto* su unione segue per insiemi $([a, b] \odot (\cdot))$ ma richiede la formula completa per intervalli con segni misti (cfr. §??).

10.9 Estensioni: segni, continui e densità

10.9.1 Estremi negativi

Per intervalli con estremi in $\mathbb{Z}$, definire:

$$[a, b] \odot [c, d] = [\min\{ac, ad, bc, bd\}, \max\{ac, ad, bc, bd\}],$$

e mantenere la definizione di somma di Minkowski. ∞_n resta definito per $n \geq 1$.

10.9.2 Raffinamento sub-unitario

Si può introdurre una griglia $\delta \in \mathbb{Q}^+$ e lavorare con estremi in $\delta\mathbb{Z}$. In tal caso ∞_n occupa $[n, n+1)$ ma le operazioni possono produrre estremi multipli di δ .

10.9.3 Densità e pesi

Associare a ciascun $[a, b]$ un peso ω (intero/razionale) che conteggia la molteplicità di origine; le operazioni aggiornano i pesi per somma (somma) o convoluzione (prodotto). La FN ignora i pesi, che sono riportati nel log.

10.10 API di riferimento

Firma dei costruttori

- `Inf(n: Nat) : Obj  $\mapsto [n, n+1)$ .`
- `FromIntervals(xs: List[(Nat, Nat)], compress: Bool) : Obj (FN interna).`
- `Top() : Obj  $\mapsto \perp$ .`

Operazioni

- `add(U,V, policy)` : $\text{Obj} \equiv U \oplus V$ con FN secondo `policy`.
- `mul(U,V, policy)` : $\text{Obj} \equiv U \odot V$ con FN.
- `scale(k,U, policy)` : $\text{Obj} \equiv k \cdot U$ con FN.
- `union(U,V, policy)` : Obj ; `inter(U,V, policy)` : Obj .
- `isSubset(U,V)` : Bool ; `equals(U,V, policy)`: Bool .

Plug-in (R, Φ)

- `registerR(name, Rfun); registerPhi(name, Phifun).`
- `regularize(U, Rname)` : Sym ; `evaluate(Sym, Phiname)` : Value .

10.11 Test di conformità

10.11.1 Casistiche minime

1. **Somma base:** $\infty_1 \oplus \infty_2 \rightarrow [3, 5)$.
2. **Prodotto base:** $\infty_1 \odot \infty_3 \rightarrow [3, 8)$.
3. **Scalatura:** $2 \cdot \infty_4 \rightarrow [8, 10)$.
4. **Union FN:** $[1, 2) \cup [2, 3)$ con `compress=true` $\rightarrow [1, 3)$; con `compress=false` $\rightarrow [1, 2) \cup [2, 3)$.
5. **Top assorbente:** $\infty_1 \oplus \perp \rightarrow \perp$, $\infty_2 \odot \perp \rightarrow \perp$.
6. **Intersezione:** $[1, 4) \cap [3, 5) = [3, 4)$.

10.11.2 Proprietà da verificare

- Associatività/commutatività di $\oplus$ (confronto liste FN).
- Monotonia rispetto a $\subseteq$.
- Idempotenza dell'FN: $\text{FN}(\text{FN}(U)) = \text{FN}(U)$.
- Invarianza rispetto alla ri-associazione: $\text{FN}((U \oplus V) \oplus W) = \text{FN}(U \oplus (V \oplus W))$.

10.12 Esempi implementativi end-to-end

10.12.1 Somma e prodotto con log

Esempio 10.1. Input: $U = \infty_1 \cup \infty_3$, $V = \infty_2$.

Somma: $U \oplus V = [1, 2) \oplus [2, 3) \cup [3, 4) \oplus [2, 3) = [3, 5) \cup [5, 7)$. Con `compress=true` $\rightarrow [3, 7)$.

Prodotto: $U \odot V = [1, 2) \odot [2, 3) \cup [3, 4) \odot [2, 3) = [2, 6) \cup [6, 12)$. Con `compress=true` $\rightarrow [2, 12)$.

10.12.2 Applicazione di (R, Φ)

Esempio 10.2. Con $R = R^{(c)}$ (centroide) e $\Phi = \Phi^{(c)}$: $R([2, 12)) = S_{2,12}^{(c)}$, $\Phi(S_{2,12}^{(c)}) = (2 + 12 - 1)/2 = 6.5$.

10.13 Esercizi e soluzioni

Esercizi

1. Calcolare in FN (con `compress=false`): $(\infty_1 \oplus \infty_2) \cup (\infty_2 \oplus \infty_3)$.
2. Con `compress=true`, valutare $(\infty_1 \cup \infty_2) \odot (\infty_3 \cup \infty_4)$.
3. Dimostrare che FN è idempotente.
4. Estendere la definizione di $\odot$ a intervalli con estremi in $\mathbb{Z}$ e verificare su $[-1, 2) \odot [3, 5)$ il risultato previsto.
5. Definire un R a soglia K e una Φ che restituisca il bordo superiore troncato. Applicarlo a $[5, 11)$ con $K = 9$.

Soluzioni

1. $\infty_1 \oplus \infty_2 = [3, 5)$; $\infty_2 \oplus \infty_3 = [5, 7)$. Con `compress=false`: $[3, 5) \cup [5, 7)$ (due blocchi).
2. Espansione: $[1, 2) \odot [3, 4) = [3, 8)$; $[1, 2) \odot [4, 5) = [4, 10)$; $[2, 3) \odot [3, 4) = [6, 12)$; $[2, 3) \odot [4, 5) = [8, 15)$. Unione: $[3, 8) \cup [4, 10) \cup [6, 12) \cup [8, 15) \xrightarrow{\text{compress}} [3, 15)$.
3. Per definizione: la seconda applicazione non introduce nuove fusioni oltre quelle già fatte.
4. Prodotti possibili: $(-1) \cdot 3 = -3$, $(-1) \cdot 5 = -5$, $2 \cdot 3 = 6$, $2 \cdot 5 = 10$. $\Rightarrow [-5, 10)$.
5. $R([5, 11)) = S_{5, \min(10, 9)} = S_{5, 9}$. $\Phi(S_{5, 9}) = 9$.

10.14 Riepilogo del capitolo

Riassunto

Cosa ricordare:

- AIE2 si implementa con un tipo semplice (unioni finite di intervalli) e una FN deterministica $O(m \log m)$.
- Le operazioni $\oplus, \odot$ e la scalatura sono definite sugli estremi e si estendono per unioni finite seguite da FN.
- $\perp$ è un top assorbente *intenzionale* per preservare l'affidabilità epistemica.
- Le politiche di compressione modulano il trade-off tra semplicità e tracciabilità.
- (R, Φ) è un livello plugin: separa la regolarizzazione dalle regole algebriche di base e consente audit.

Chapter 11

Verifica formale, coerenza e modelli

Riassunto

Obiettivo. Fornire una base rigorosa per AIE2: (i) un modello insiemistico canonico; (ii) prove di *chiusura*, *ben-definizione* e *coerenza* delle operazioni; (iii) proprietà algebriche (associatività, commutatività, monotonia, esistenza di top assorbente); (iv) una procedura decisionale per uguaglianza e inclusione; (v) la cornice logica per l'estensione con il livello (R, Φ) e relative garanzie (suono/monotono).

11.1 Dominio semantico canonico

11.1.1 Oggetti e forma normale

Sia $\mathcal{G} = \{\infty_n \mid n \in \mathbb{N}_{\geq 1}\}$ con $\infty_n = [n, n+1) \subset \mathbb{R}$. Definiamo l'*universo AIE2* come

$$\mathbb{A} := \left\{ U = \bigcup_{i=1}^k [a_i, b_i) \mid k \in \mathbb{N}, a_i, b_i \in \mathbb{N}, 1 \leq a_i < b_i \right\} \cup \{\perp\}.$$

Una *Forma Normale* (FN) per $U \in \mathbb{A} \setminus \{\perp\}$ è una rappresentazione

$$U \equiv \bigcup_{j=1}^m [A_j, B_j) \quad \text{con} \quad [A_j, B_j) \cap [A_{j'}, B_{j'}) = \emptyset \ (j \neq j'), \quad A_1 < \dots < A_m,$$

eventualmente con *compressione contigua* $[u, v) \cup [v, w) \mapsto [u, w)$ secondo una *politica* fissata.

11.1.2 Operazioni primitive (semantica insiemistica)

Per $U, V \in \mathbb{A} \setminus \{\perp\}$:

$$U \oplus V := \text{FN} \left(\bigcup_{[a,b] \subseteq U} \bigcup_{[c,d] \subseteq V} [a+c, b+d] \right),$$

$$U \odot V := \text{FN} \left(\bigcup_{[a,b] \subseteq U} \bigcup_{[c,d] \subseteq V} [a \cdot c, b \cdot d] \right),$$

$$k \cdot U := \text{FN} \left(\bigcup_{[a,b] \subseteq U} [ka, kb] \right) \quad (k \in \mathbb{N}).$$

Per ogni $U \in \mathbb{A}$: $U \oplus \perp = \perp$, $U \odot \perp = \perp$, $k \cdot \perp = \perp$.

Intuizione

Questo modello è “concreto”: gli oggetti sono unioni finite di intervalli semiaperti interi, e le operazioni sono somme/prodotti/scalature di Minkowski, seguite da normalizzazione (FN).

11.2 Ben-definizione, chiusura e coerenza

Proposizione 11.1 (Chiusura). *Se $U, V \in \mathbb{A} \setminus \{\perp\}$ e $k \in \mathbb{N}$, allora $U \oplus V \in \mathbb{A}$, $U \odot V \in \mathbb{A}$ e $k \cdot U \in \mathbb{A}$.*

Proof. Per definizione, U e V sono unioni finite di intervalli $[a, b]$ e $[c, d]$ con estremi interi e $a < b$, $c < d$. Le operazioni sugli estremi producono ancora estremi interi con disuguaglianze preservate ($a + c < b + d$, $ac < bd$ dato $a, b, c, d \geq 1$; per la scalatura $ka < kb$ se $k \geq 1$). L'unione finita è ancora unione finita di intervalli della stessa forma. La FN non esce da $\mathbb{A}$. I casi con $\perp$ sono per definizione in $\mathbb{A}$. $\square$

Proposizione 11.2 (Indipendenza dalla rappresentazione). *La FN è idempotente e canonica rispetto alla politica fissata. In particolare, $\text{FN}(\text{FN}(U)) = \text{FN}(U)$ e, a parità di politica, qualsiasi espansione di U produce la stessa FN.*

Proof. La procedura standard di merge di intervalli ordinati elimina sovrapposizioni e, se attivata, compatta i blocchi contigui. Rieseguire la stessa procedura non introduce nuove fusioni né scissioni; segue l'idempotenza. La canonicità discende dall'ordinamento deterministico e dalle stesse regole di fusione della politica scelta. $\square$

Proposizione 11.3 (Coerenza del top assorbente). *Per ogni $U \in \mathbb{A}$: $U \oplus \perp = \perp$ e $U \odot \perp = \perp$. Inoltre $k \cdot \perp = \perp$.*

Proof. Definizione: $\perp$ rappresenta informazione *non affidabile*; la sua propagazione garantisce suono epistemico. Non si introducono contraddizioni, poiché $\perp$ non viene mai identificato con unione di intervalli. $\square$

[Assenza di contraddizioni] Non esiste U tale che $U = \perp$ nella semantica insiemistica; dunque il sistema è coerente (consistente) rispetto al modello.

11.3 Proprietà algebriche

Nel seguito assumiamo $U, V, W \in \mathbb{A} \setminus \{\perp\}$ e $k, \ell \in \mathbb{N}$. Le uguaglianze sono da intendersi *modulo FN*.

Proposizione 11.4 (Commutatività e associatività di $\oplus$). $(U \oplus V) = (V \oplus U)$ e $(U \oplus V) \oplus W = U \oplus (V \oplus W)$.

Proof. La somma di Minkowski è commutativa e associativa sugli insiemi di $\mathbb{R}$. La FN preserva l'uguaglianza insiemistica. $\square$

Proposizione 11.5 (Associatività del prodotto $\odot$ e commutatività). $(U \odot V) \odot W = U \odot (V \odot W)$ e $U \odot V = V \odot U$.

Proof. Il prodotto di Minkowski di sottoinsiemi di $\mathbb{R}_{\geq 1}$ con estremi interi è commutativo e associativo: $[a, b] \odot [c, d] = [ac, bd]$ e il composto produce gli stessi estremi indipendentemente dall'ordine. $\square$

Proposizione 11.6 (Distributività su unioni finite). $U \odot (V \cup W) = (U \odot V) \cup (U \odot W)$ e $U \oplus (V \cup W) = (U \oplus V) \cup (U \oplus W)$, *poi FN*.

Proposizione 11.7 (Monotonia). Se $U \subseteq V$ allora $U \oplus W \subseteq V \oplus W$ e $U \odot W \subseteq V \odot W$; inoltre $k \cdot U \subseteq k \cdot V$.

Proposizione 11.8 (Ordine e top). Con l'ordine per inclusione $(\mathbb{A} \setminus \{\perp\}, \subseteq)$ e $\perp$ come elemento top esterno, si ottiene un poset completo per unioni finite; $\perp$ è assorbente per $\oplus, \odot$.

Intuizione

Algebricamente, la parte “regolare” $\mathbb{A} \setminus \{\perp\}$ è una famiglia di sottoinsiemi chiusa per somma/prodotto di Minkowski e scalatura; $\perp$ aggiunge un *top epistemico* che preserva la suonità operativa.

11.4 Decidibilità di uguaglianza e inclusione

Proposizione 11.9 (Uguaglianza). Sia fissata la politica di FN. Allora $U = V$ sse le liste FN coincidono elemento per elemento (stessi intervalli nello stesso ordine).

Proof. La FN è canonica a parità di politica (Prop. ??). Due insiemi uguali hanno la stessa decomposizione disgiunta canonica. $\square$

Proposizione 11.10 (Inclusione). *Dato $U = \bigcup_i [A_i, B_i]$ e $V = \bigcup_j [C_j, D_j]$ in FN, $U \subseteq V$ sse per ogni $[A_i, B_i]$ esiste una famiglia di indici J_i tale che $[A_i, B_i] \subseteq \bigcup_{j \in J_i} [C_j, D_j]$. Il test è decidibile in tempo $O(m + n)$ dopo l'ordinamento (già effettuato in FN).*

Proof. Riduzione a sweep line su segmenti disgiunti e ordinati. $\square$

[Complessità della FN e delle verifiche] La FN di un multinsieme di t intervalli si calcola in $O(t \log t)$. I test di uguaglianza/inclusione post-FN sono $O(m + n)$.

11.5 Modello astratto e completezza equazionale

Definizione 11.1 (Calcolo equazionale AIE2). È il sistema di riscrittura che comprende: regole di espansione ($\oplus, \odot, k \cdot$) su intervalli elementari, riscritture di FN (merge/sort/compress) e propagazione di $\perp$.

Teorema 11.11 (SuonoAIEblue¹). *Ogni identità derivabile nel calcolo equazionale AIE2 è vera nel modello insiemistico di §??.*

Idea della dimostrazione. Le regole primitive corrispondono a operazioni di Minkowski, per cui valgono le proprietà di §??. La FN è una normalizzazione confluenti/idempotente (Prop. ??); la propagazione di $\perp$ è definizionale. $\square$

Teorema 11.12 (Completezza equazionale (frammento positivo)). *Fissata la politica di FN, per espressioni costruite su intervalli $[n, n + 1]$ con $n \geq 1$, $\oplus, \odot, k \cdot$ e unione finita, il calcolo equazionale AIE2 è completo rispetto al modello: se due espressioni denotano lo stesso insieme, sono inter-dimostrabili nel sistema.*

Schema. Si applica la normalizzazione strutturale fino a FN; due oggetti denotano lo stesso insieme se e solo se hanno la stessa FN (Prop. ??). Il sistema resta nei confini di identità insiemistiche più regole di normalizzazione deterministiche. $\square$

Attenzione

La completezza logica per linguaggi più ricchi (pesi, segni misti, densità) richiede estensioni tecniche (formula generale del prodotto d'intervalli con segni, *weights* come annotazioni, ecc.).

¹“Suono” rispetto al modello insiemistico canonico.

11.6 Livello di regolarizzazione (R, Φ)

Definizione 11.2 (Interfacce astratte). Siano $(\mathbb{A}, \preceq)$ il poset con $\perp$ top esterno e $\preceq$ l'ordine per inclusione esteso con $U \preceq \perp$. Un *regolarizzatore* è una funzione totale $R : \mathbb{A} \rightarrow \mathbb{S}$ in un codominio $\mathbb{S}$, e un *funzionale* $\Phi : \mathbb{S} \rightarrow X$ (tipicamente $X \subseteq \mathbb{R}$).

Definizione 11.3 (Regolarizzatori monotoni). Diremo che R è *monotono* se $U \preceq V \Rightarrow R(U) \preceq_{\mathbb{S}} R(V)$ per un ordine prefissato su $\mathbb{S}$. Analogamente, Φ è monotono se $S \preceq_{\mathbb{S}} T \Rightarrow \Phi(S) \leq_X \Phi(T)$.

Proposizione 11.13 (Suono metodologico). Se R e Φ sono monotoni e compatibili con FN ($R(\text{FN}(U)) = R(U)$), allora l'intera pipeline $U \mapsto \Phi(R(U))$ è stabile rispetto all'informazione: raffinare U (unione più ampia) non può produrre un valore “contraddittorio” rispetto all'ordine.

Intuizione

(R, Φ) è un *livello esterno* che non altera l'algebra interna: la rende solo “numerica” quando serve, mantenendo tracciabilità e coerenza con l'ordine d'informazione.

11.7 Estensioni del modello

Segni misti. Se si ammettono estremi interi in $\mathbb{Z}$, per il prodotto si usa $[a, b] \odot [c, d] = [\min\{ac, ad, bc, bd\}, \max\{ac, ad, bc, bd\}]$. Il resto della teoria si adatta in modo standard; la FN resta idempotente/canonica.

Griglia sub-unitaria. Con griglia $\delta \in \mathbb{Q}^+$, gli estremi vivono in $\delta\mathbb{Z}$. ∞_n resta $[n, n+1)$, ma le operazioni producono estremi multipli di δ . Le prove si trasferiscono mutatis mutandis.

Annotazioni di peso. Associare pesi $\omega \in \mathbb{N}$ a ciascun blocco $[a, b)$ permette conteggi di molteplicità/origine. Le proprietà insiemistiche restano, i pesi seguono somma (per $\oplus$) e combinazione tipo convoluzione (per $\odot$).

11.8 Esempi verificati e micro-prove

Esempio 11.1 (Associatività operativa di $\oplus$). Siano $U = [a, b)$, $V = [c, d)$, $W = [e, f)$. $(U \oplus V) \oplus W = [a + c, b + d) \oplus [e, f) = [a + c + e, b + d + f)$. Simmetricamente $U \oplus (V \oplus W) = [a + e + c, b + f + d)$. Gli estremi coincidenti danno uguaglianza modulo FN.

Esempio 11.2 (Monotonia del prodotto). Se $U \subseteq V$ allora per ogni blocco $[x, y) \subseteq U$ esiste $[X, Y) \subseteq V$ con $[x, y) \subseteq [X, Y)$. Per ogni $[c, d) \subseteq W$, $[x, y) \odot [c, d) \subseteq [X, Y) \odot [c, d)$. Unione su tutti i blocchi $\Rightarrow U \odot W \subseteq V \odot W$.

Esempio 11.3 (Procedura decisionale). Dato $U = [1, 2) \cup [4, 6)$ e $V = [1, 3) \cup [4, 5)$ in FN: $[1, 2) \subseteq [1, 3)$ e $[4, 6) \not\subseteq [4, 5)$ (bordo superiore $6 > 5$). Dunque $U \not\subseteq V$. L'uguaglianza fallisce per blocchi diversi.

11.9 Esercizi con soluzioni

Esercizi

1. Dimostrare che la FN è idempotente per qualsiasi politica deterministica di compressione dei blocchi contigui.
2. (Decidibilità) Progettare un algoritmo lineare (post-FN) per verificare $U \subseteq V$, e analizzarne la complessità.
3. (Distributività) Verificare su blocchi elementari che $U \odot (V \cup W) = (U \odot V) \cup (U \odot W)$.
4. (Top) Mostrare che non esiste sequenza di riscritture che trasformi $\perp$ in una unione finita di intervalli.
5. (Segni misti) Con estremi in $\mathbb{Z}$, calcolare $[-1, 2) \odot [3, 5)$ e verificare le proprietà di monotonia e associatività.

Soluzioni

1. La procedura FN ordina e fonde; una seconda applicazione non trova nuovi blocchi contigui/sovrapposti: segue l'idempotenza.
2. Si scansionano in parallelo le liste ordinate; ogni blocco di U deve essere coperto da blocchi di V . Avanzamento a due puntatori $\Rightarrow O(m + n)$.
3. Per blocchi $[a, b)$, $[c, d)$, $[e, f)$ si ha $[a, b) \odot ([c, d) \cup [e, f)) = [ac, bd) \cup [ae, bf)$, come desiderato.
4. La riscrittura con $\perp$ è solo assorbente; non esiste regola inversa che “crei” intervalli da $\perp$.
5. Prodotti estremi: $(-1) \cdot 3 = -3$, $(-1) \cdot 5 = -5$, $2 \cdot 3 = 6$, $2 \cdot 5 = 10$. Quindi $[-5, 10)$. Monotonia/associatività seguono dalla formula generale.

11.10 Riepilogo del capitolo

Riassunto

Cosa abbiamo stabilito:

- Un modello insiemistico concreto per AIE2 rende *suono* il calcolo operativo: chiusura, ben-definizione e coerenza.
- La FN è idempotente e canonica (a parità di politica), consentendo procedure decisionali efficienti.
- $\oplus, \odot, k\cdot$ ereditano commutatività/associatività/monotonia dalle corrispondenti operazioni di Minkowski.
- $\perp$ è un top assorbente che preserva la suonità epistemica senza introdurre contraddizioni.
- Il livello (R, Φ) è esterno e, se monotono/compatibile con FN, garantisce stabilità informativa dei valori prodotti.

Appendici

Chapter 12

Dimostrazioni tecniche complete

12.1 Dimostrazione completa: Somma su generatori

Teorema 12.1. Per $m, n \geq 1$:

$$\infty_m \oplus \infty_n = [m+n, m+n+2) = \infty_{m+n} \cup \infty_{m+n+1}.$$

Proof. Sia $A = [m, m+1)$, $B = [n, n+1)$. Per ogni $a \in A$, $b \in B$ si ha $m \leq a < m+1$ e $n \leq b < n+1$. Sommandole: $m+n \leq a+b < m+n+2$. Quindi $A \oplus B \subseteq [m+n, m+n+2)$.

Viceversa, per ogni x con $m+n \leq x < m+n+2$ bisogna mostrare che esistono $a \in A$, $b \in B$ tali che $a+b = x$. Sia ad esempio $a = \max(m, x-(n+1)+\varepsilon)$ con ε piccola positiva; esiste un $b = x-a$ che cade in $[n, n+1)$. Più elementare: se $m+n \leq x < m+n+1$ prendi $a = m$ e $b = x-a \in [n, n+1)$; se $m+n+1 \leq x < m+n+2$ prendi $a = m+1-\delta$ piccolo e $b = x-a$. Formalmente, usando densità dei reali, esistono tali a, b . Quindi l'inclusione opposta vale e l'uguaglianza insiemistica è stabilita.

Infine, dividendo l'intervallo $[m+n, m+n+2)$ in blocchi unitari:

$$[m+n, m+n+2) = [m+n, m+n+1) \cup [m+n+1, m+n+2) = \infty_{m+n} \cup \infty_{m+n+1}.$$

□

(Analoghe dimostrazioni formali si ottengono per produtt o scalatura; in ambiente formale i passaggi usano `Nat.add_lt_add`, `Nat.mul_lt_mul...`)

12.2 Unicità della forma normale

Teorema 12.2. Ogni $U \in \mathcal{U}$ (unione finita di intervalli consecutivi) ha una rappresentazione unica come $[p, q+1)$ con $p \leq q$.

Proof. Se $U = \bigcup_{k=a}^b \infty_k = [a, b+1)$ e anche $U = [a', b'+1)$ allora gli estremi minimi coincidono: il minimo di U è a e anche a' , quindi $a = a'$; analogo argomento per il massimo (uguaglianza di estremi) conclude. $\square$

Chapter 13

Tavole riassuntive e cheat-sheet

Formule essenziali (ripasso rapido)

- Generatore: $\infty_n = [n, n + 1)$.
- Somma: $[a, b) \oplus [c, d) = [a + c, b + d)$.
- Prodotto: $[a, b) \odot [c, d) = [ac, bd)$ (per estremi positivi); formula completa coi segni: $[\min\{ac, ad, bc, bd\}, \max\{ac, ad, bc, bd\})$.
- Scalatura: $k \cdot [a, b) = [ka, kb)$.
- FN: $[p, q)$ rappresenta $\bigcup_{k=p}^{q-1} \infty_k$.
- Indeterminato: $\frac{0}{0} = \perp$ con propagazione forte.

Checklist rapida per implementatori

1. usare smart constructors che mantengono invarianti ($\text{start} < \text{stop}$);
2. implementare e testare `fn_merge` con molteplici casi limite (sovrapposizioni, contiguità, vuoto);
3. marcare $\perp$ come top e implementare propagazione forte come caso base;
4. scrivere property-based tests (se possibile) che verificano le identità algebriche su campioni random.

Chapter 14

Bibliografia e letture consigliate

Bibliography

- [1] A. Robinson. *Non-standard Analysis*. North-Holland, 1966.
- [2] L. Ahlfors. *Complex Analysis*. McGraw-Hill, 1979.
- [3] C. Itzykson and J.-B. Zuber. *Quantum Field Theory*. McGraw-Hill, 1980.
- [4] B. Chellas. *Modal Logic: An Introduction*. Cambridge Univ. Press, 1980.
- [5] I. Stewart. *Galois Theory*. Chapman and Hall, 2004.
- [6] The Lean community. *mathlib4 (online)*. (vedi documentazione ufficiale per le librerie matematiche in Lean 4).

Chapter 15

Conclusioni e prospettive future

Riassunto

Scopo: questo capitolo conclude il percorso intrapreso nel volume. Racogliamo i risultati principali della teoria AIE2, valutiamo i punti di forza, riconosciamo i limiti aperti e indichiamo direzioni per future ricerche e applicazioni.

15.1 Sintesi dei risultati principali

1. **Fondazione assiomatica:** abbiamo costruito un sistema coerente di regole operative che gestisce infiniti etichettati, intervalli e indeterminati, in un quadro rigoroso ma intuitivo.
2. **Chiusura e forma normale:** ogni operazione algebrica tra infiniti etichettati ammette una rappresentazione canonica (FN), che garantisce semplicità, univocità e decidibilità.
3. **Funzione di valutazione:** l'introduzione della coppia (R, Φ) ha reso possibile collegare AIE2 con campi numerici classici e con procedure di regolarizzazione usate in fisica e matematica applicata.
4. **Confronto con altre teorie:** abbiamo messo in luce le differenze e le complementarità tra AIE2, Analisi Non Standard, logica modale e teoria di Galois, mostrando come AIE2 offra una prospettiva inedita e compatibile.
5. **Implementabilità:** la formalizzazione in Lean e Coq ha dimostrato che la teoria non è solo un insieme di intuizioni, ma un sistema meccanizzabile e quindi adatto a proof assistants e a software di calcolo simbolico.
6. **Applicazioni:** AIE2 è stata discussa in contesti di matematica pura, fisica teorica, informatica ed economia, mostrando una flessibilità che ne giustifica l'interesse interdisciplinare.

15.2 Punti di forza di AIE2

- **Semplicità concettuale:** le regole operative si fondano su concetti insiemistici elementari (unioni, somme, prodotti).
- **Tracciabilità:** ogni infinito conserva un'etichetta che ne testimonia l'origine, caratteristica assente in molte altre teorie.
- **Flessibilità:** la possibilità di introdurre diversi regolatori (R, Φ) rende AIE2 adattabile a più domini.
- **Compatibilità:** AIE2 non esclude le teorie esistenti ma dialoga con esse, fornendo strumenti integrativi.
- **Implementabilità:** grazie alla formalizzazione, AIE2 non resta confinata nel piano teorico ma diventa sperimentabile in software e librerie matematiche.

15.3 Limiti e questioni aperte

Nonostante i risultati raggiunti, AIE2 presenta ancora sfide:

1. **Scelta del regolatore (R, Φ) :** mancano criteri universali che guidino la scelta della funzione di regolarizzazione più adatta in un dato contesto.
2. **Gestione dell'indeterminato:** la propagazione forte di $\perp$ è una scelta prudente ma potrebbe risultare troppo restrittiva in applicazioni specifiche.
3. **Generalizzazione:** estendere AIE2 oltre gli intervalli unitari, ad esempio ad intervalli reali arbitrari o a strutture più complesse (reti, grafi), è un obiettivo ancora da sviluppare.
4. **Didattica e divulgazione:** AIE2 richiede un linguaggio adeguato per essere introdotta a studenti e ricercatori non specialisti, evitando tecnicismi eccessivi.

15.4 Prospettive future

Le direzioni principali per lo sviluppo futuro includono:

- **Approfondimento teorico:** estendere la teoria a contesti algebrici più generali (anelli, campi, categorie) e indagare la compatibilità con la teoria degli ordini e la logica modale.

- **Fisica teorica:** sviluppare procedure standard di regolarizzazione basate su AIE2 per la teoria dei campi, la gravità quantistica e la termodinamica dei buchi neri.
- **Informatica:** creare librerie dedicate (in Lean, Coq, Haskell, Python) che consentano a ricercatori e studenti di sperimentare direttamente con gli oggetti AIE2.
- **Economia e scienze sociali:** testare modelli AIE2 in sistemi complessi (dinamiche di crescita, equilibri instabili, teoria dei giochi con divergenze).
- **Filosofia della scienza:** utilizzare AIE2 come caso di studio per analizzare il rapporto tra convenzioni matematiche, strumenti computazionali e interpretazioni epistemiche.

15.5 Considerazioni finali

Riassunto

AIE2 non pretende di sostituire i grandi sistemi matematici già esistenti, ma di proporsi come un *framework operativo* per affrontare l'infinito e l'indeterminato in modo più tracciabile, calcolabile e applicabile.

La teoria è giovane, ma ha già dimostrato:

- coerenza formale;
- capacità di dialogo con altre aree della matematica;
- potenzialità di applicazione in fisica, informatica ed economia;
- apertura a riflessioni filosofiche e didattiche.

Il cammino futuro dipende ora dalla comunità scientifica: il valore di AIE2 non sta solo nelle regole che abbiamo codificato, ma soprattutto nell'uso che i ricercatori sapranno farne.

Postfazione

Riassunto

Questa postfazione è pensata come messaggio diretto al lettore. Non riassume soltanto i contenuti del volume, ma suggerisce un percorso concreto per avvicinarsi allo studio e all'applicazione della teoria AIE2.

Per lo studente

Se sei uno studente universitario o un giovane ricercatore, il consiglio principale è:

- **Inizia dagli esempi:** prova a calcolare a mano espressioni come $\frac{1}{0}, \frac{2}{0}, \frac{0}{0}$, o somme e prodotti di ∞_n . Questo ti aiuterà a familiarizzare con le regole operative.
- **Costruisci la Forma Normale (FN):** esercitati a trasformare unione di intervalli in FN, perché è la chiave per comprendere la struttura di AIE2.
- **Confronta con altre teorie:** riprendi un esempio che conosci (per esempio una serie divergente) e confronta come viene trattato in AIE2, in analisi classica e in analisi non standard.
- **Usa gli strumenti formali:** se conosci Lean o Coq, prova ad implementare le definizioni più semplici (come l'addizione di due intervalli). Anche pochi esempi già rafforzano l'intuizione.

Per il ricercatore

Se sei un ricercatore, AIE2 può offrirti nuove prospettive operative:

- **Fisica teorica:** usa AIE2 per gestire divergenze nei calcoli (catastrofe ultravioletta, entropia dei buchi neri) con regolarizzazioni tracciabili.
- **Informatica:** integra AIE2 in software simbolico o librerie Lean/Coq per costruire strumenti sperimentali. Le definizioni presentate sono già pronte per una codifica.

- **Matematica pura:** esplora l'interazione tra AIE2 e logica modale, teoria degli ordini e teoria dei campi. Ogni etichetta infinita può essere vista come un “mondo possibile” o uno “strato”.
- **Scienze sociali ed economia:** modella sistemi che esplodono (dinamiche di crescita non lineare, crisi, instabilità) con operatori AIE2, che mantengono l'informazione della divergenza senza perderla in un unico simbolo di infinito.

Suggerimenti di lettura e lavoro

1. Rileggi le **tabelle riassuntive** in Appendice: sono la mappa più rapida per orientarsi.
2. Consulta i testi in bibliografia, soprattutto Robinson per l'Analisi Non Standard, Ahlfors per l'analisi complessa, e Itzykson-Zuber per la fisica dei campi.
3. Prova ad applicare AIE2 ad un problema che già conosci bene: è il modo migliore per capire se il framework aggiunge valore pratico.
4. Condividi i risultati: AIE2 è pensata per essere esplorata, criticata, adattata. La crescita della teoria dipenderà dal confronto con altri ricercatori.

Considerazione finale

AIE2 è una teoria giovane, ancora in costruzione, ma con solide fondamenta. Abbiamo voluto presentarla in modo rigoroso e accessibile, convinti che un nuovo linguaggio sull'infinito e sull'indeterminato possa stimolare nuove idee, non solo in matematica ma in ogni disciplina che incontra limiti, divergenze e complessità.

*Non è l'infinito a essere paradossale: è il nostro modo di guardarlo
che ne decide la forma.*

— Salvatore Crapanzano

[language=lean, caption=Versione estesa di AIE2 in Lean con Forma Normale, label=code:lean_fn]import data.finset.basic

– Un intervallo $[a,b)$ con $a, b \in \mathbb{R}$ e $a < b$ structure Interval := (a b : ℝ) (h : a < b)

– Una Forma Normale (FN) è un insieme finito di intervalli @[derive decidable_eq]def FN := finset Interval

– Indeterminato inductive AIE2 | $\text{fn} : \text{FN} \rightarrow \text{AIE2}$ – unione finita di intervalli
 | $\text{indet} : \text{AIE2}$ – indeterminato

open AIE2

– Somma Minkowskiana di due intervalli: $[a,b) \cup [c,d) = [a+c, b+d)$ def
 $\text{interval}_a \text{dd}(IJ : \text{Interval}) : \text{Interval} := a := I.a + J.a, b := I.b + J.b, h := \text{add}_l t_a \text{dd} I.h J.h$

– Prodotto di due intervalli (solo caso positivo per semplicità): $-[a,b) \cup [c,d) =$
 $[a*c, b*d)$ def $\text{interval}_m \text{ul}(IJ : \text{Interval}) : \text{Interval} := a := I.a * J.a, b := I.b * J.b, h := \text{mult}_m \text{ul} I.h J.h$

– Estensione della somma a due FN def $\text{fn}_a \text{dd}(UV : \text{FN}) : \text{FN} := U.\text{bind}(I, V.\text{image}(J, \text{interval}_a \text{dd} I J))$

– Estensione del prodotto a due FN def $\text{fn}_m \text{ul}(UV : \text{FN}) : \text{FN} := U.\text{bind}(I, V.\text{image}(J, \text{interval}_m \text{ul} I J))$

– Operazioni su AIE2 def $\text{add} : \text{AIE2} \rightarrow \text{AIE2} \rightarrow \text{AIE2}$ | $(\text{fn } U) (\text{fn } V) := \text{fn}$
 $(\text{fn}_a \text{dd} UV) | \text{indet}_i := \text{indet}_i | \text{indet} := \text{indet}$

def $\text{mul} : \text{AIE2} \rightarrow \text{AIE2} \rightarrow \text{AIE2}$ | $(\text{fn } U) (\text{fn } V) := \text{fn } (\text{fn}_m \text{ul} UV) | \text{indet}_i =$
 $\text{indet}_i | \text{indet} := \text{indet}$

– Esempio: definizione di un singolo infinito etichettato $_n = [n, n+1)$ def $\text{inf}(n :$
 $) : \text{AIE2} := \text{fn}(\text{Interval.mk } n(n+1)(\text{nat.lt}_s \text{ucc}_s \text{el } \text{fn})) : \text{FN}$

– Test eval $\text{add}(\text{inf } 1) (\text{inf } 2)$ – dovrebbe rappresentare $[3,5)$ cioè $_{34} \text{eval mul}(\text{inf } 2)(\text{inf } 3) -$
 – $\text{dovrebbe rappresentare } [6, 12) =_6 \dots_1 1$